

P2P sisteminin RFC Belgesi:

CEM ÇELEBİ11401600

1. Tanımlar

- PEER: Kullanıcıların görüntü işleme sistemine dahil olabilmesi için kullanacakları (eş) uç yazılım.
- NEGOTIATOR: Bir eşin diğer eş bilgilerini çekebileceği ve kendisini kaydedeceği sunucu.
- PEER CLIENT: PEER'in istek gönderen tarafı.
- PEER SERVER: PEER'in bağlantı dinleyici tarafı.
- PEER IP: Eşin başka bir eşten bağlantı bekleyeceği IP adresi.
- PEER PORT: Eşin başka bir eşten bağlantı bekleyeceği port numarası.
- CONNECT POINT: Bir eşin diğer eşlerin bağlanabilmesi için NEGOTIATOR'a göndereceği ve dinliyor olacağı [PEER IP, PEER PORT] ikilisi.
- CONNECT POINT LIST: Bir eş veya sunucunun bildiği CONNECT POINT listesi.
- FUNCTIONS: PEER sistemindeki paylaşılan dosyaların bulunduğu dizin.
- UPDATE INTERVAL: CONNECT POINT LIST'in güncellenme aralığı.
- CHUNKSIZE: Görüntü parçası boyutu. Kare matris olacağı için tek bir sayıyeterli.
- REGISTER: Eşin sunucuya bağlanıp kendisini kaydetmesi.
- CHUNK: Paylaşılan görüntü matrisi.

2. Sistem Genelindeki Gereksinimler:

2.1. Kullanım gereksinimleri:

- 2.1.1. Bu sistem, yerel ağlarda çalışabilecek bir P2P (peer to peer/ eşten eşe) görüntü işleme sistemidir.
- 2.1.2. Bu sistem TCP kullanmaktadır.
- 2.1.3. Bir kullanıcı sistemden görüntü işleme isteği yaparak faydalanabilir, bunun için:
 - 2.1.3.1. O kullanıcının IP numarasını NEGOTIATOR'a eklemesi gerekmektedir.
 - 2.1.3.2. O kullanıcının Port numarasını NEGOTIATOR'a eklemesi gerekmektedir.
 - 2.1.3.3. Sistem dahilinde kullanıcı adı ve parola yoktur.

Bu belge bundan sonra ana hatlarıyla Sunucu ve Eşler Arası Gereksinimler olarak ikiye ayrılmaktadır. İlkinde Sunucu tarafından bahsedilmiştir, ikincisinde Eşler Arası Gereksinimler'den bahsedilmiştir. Bu her iki kısım da Genel ve Teknik olarak ikiye ayrılmaktadır. Genel kısmında bahsedilen X.Y.Z maddesinin teknik gereksinimleri Teknik kısmında X.Y+1.Z bahsedilmiştir.

3. Sunucunun Gereksinimleri:

3.1. Sunucunun Genel Gereksinimleri:

- 3.1.1. İlk kez sisteme bağlanacak bir eşin IP ve Port numaralarını alıp CONNECTION_POINT_LIST isimli değişkene eklemesi gerekmektedir.
- 3.1.2. Bunun akabinde, sunucu yeni bağlanan eşe tekrar bağlanmayı deneyip, bir sıkıntı olmadığını teyit edecektir.
- 3.1.3. Üstteki adımın tamamlanması akabinde, sunucu, CONNECTION_POINT_LIST içerisindeki tüm CONNECTION_POINT'leri yeni bağlanan eşe gönderecektir.
- 3.1.4. UPDATE_INTERVAL süresi doldukça, sunucu, CONNECTION_POINT_LIST'teki tüm CONNECTION_POINT'leri dolaşp, onlara CONNECTION_POINT_LIST'in içeriğini gönderecektir.
- 3.1.5. Sunucu belirli zaman aralıklarında kendi üzerindeki CONNECTION_POINT_LIST'i güncellemesi gerekmektedir ????
- 3.1.6. Görüntü verisi alışverişi sunucu üzerinden yapılmayacaktır.

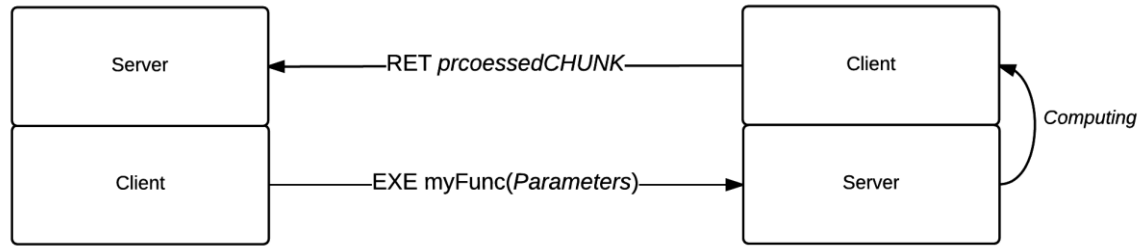
3.2. Sunucunun Teknik Gereksinimleri:

- 3.2.1.
 - 3.2.1.1. Sunucu, eşten alacağı CONNECTION_POINT bilgisi şu şekilde olmalıdır: *“USR XXX.XXX.XXX.XXX/PPPPP”*.
XXX.XXX.XXX.XXX: Eşin IP numarası.
PPPPPP: Eşin Port numarası.
 - 3.2.1.2. Sunucunun CONNECTION_POINT isimli değişkeni *Tuple* tipinde olacaktır.
 - 3.2.1.3. Sunucunun CONNECTION_POINT_LIST isimli değişkeni *Dictionary* tipinde olacaktır. *Key*'si CONNECTION_POINT olacaktır, ve *Value*'su *“lastseen on dd/mm/yyyyHH:MM:SS UTC”* olacaktır.
- 3.2.2.
 - 3.2.2.1. Yeni bağlanan eşe tekrar bağlanma işlemi sunucunun eşe *“TIC”* yollayıp *“TOC”* almasıyla tamamlanacak. *“TOC”* cevabı alınamazsa, bağlantıyı kesip, CONNECTION_POINT_LIST'ten silecektir.
- 3.2.3.
 - 3.2.3.1. Sunucu, eşlere *“CFSXXX.XXX.XXX.XXX/PPPPP”* formatında gönderecektir kendi CONNECTION_POIN_LIST içeriğinin.
XXX.XXX.XXX.XXX: Eşin IP numarası.
PPPPPP: Eşin Port numarası.
- 3.2.4.
 - 3.2.4.1. UPDATE_INTERVAL'in seçimi 5 dakika (300 saniye) olarak alınmıştır.
 - 3.2.4.2. Sunucu, eşlere *“CFS XXX.XXX.XXX.XXX/PPPPP”* formatında gönderecektir kendi CONNECTION_POIN_LIST içeriğinin.
XXX.XXX.XXX.XXX: Eşin IP numarası.
PPPPPP: Eşin Port numarası.
CFS: ConnectionPointList from Server

4. Eşler Arası Gereksinimler:

4.1. Eşler Arası Genel Gereksinimler:

- 4.1.1. Bir eş iki parçadan oluşur: client ve server tarafları.
- 4.1.2. Bir eş client tarafından istekte bulunur.
- 4.1.3. Bir eş server tarafından kendisine bulunulan isteklere cevap verir.
- 4.1.4. Bir eş, bir eşe elindeki fonksiyonları sorabilir.
- 4.1.5. Aynı şekilde, her fonksiyonun parametresini sorabilir.
- 4.1.6. CONNECTION_POINT_LIST sorgusu yapılabilir.
- 4.1.7. Eşler Execute isteği yapabilir.
- 4.1.8. Execute işlemi şu şekilde gerçekleşmektedir: P1 işlenmesi istediği resmi CHUNK'lara böler, her birini P2'ye fonksiyonu ve parametreleriyle yollar, ve bunun cevabını CHUNK'lar olarak geri alır. Bu işlemin detayları şu şekildedir: EXE isteğinden sonra, eğer P2 CHUNK'ı düzgün biçimde ele geçirmişse, EXOK mesajı döndürüp CHUNK'ı sağlam bir biçimde aldığını P1'e belirtir.
- 4.1.9. P2'nin P1'e RET gönderme işleminin detayları ise şu şekildedir: P2 CHUNK'ı işlemeyi bitirdiğinde, P1'e RETOK mesajı atar ve P1, P2'den CHUNK'ı RET komutuyla ister, P1 CHUNK'ı RET CHUNK formatında alır.



Şekil 1: P1 ve P2 arasındaki bilgi alışverişinin genel mantığı.

4.2. Eşler Arasında Teknik Gereksinimler:

- 4.2.1.
 - 4.2.1.1. P1 P2'den fonksiyon bilgilerini isterken şu şekilde isteyecektir: "FUN"
 - 4.2.1.2. P2 buna cevap olarak şu şekilde cevap verecektir: "RFUN Fun1,Fun2,Fun3...Funn"
- 4.2.2.
 - 4.2.2.1. P1 P2'den istediği fonksiyonun parametrelerini şu şekilde isteyecektir: "PAR myFun"
 - 4.2.2.2. P2 buna cevap olarak şu şekilde cevap verecektir: "RPAR Par1,Par2,Par3...Parn"
- 4.2.3.
 - 4.2.3.1. Teknik bakımdan CONNECTION_POINT_LIST haberleşmesi tıpkı 3.2.1.1 bölümündeki gibi yapılmaktadır, ekstra eklenecek bir şey yok.
- 4.2.4.
 - 4.2.4.1. P1 P2'den Execute işlemini şu şekilde isteyecektir: "EXE myFun(Par1,Par2,Par3...Parn)"
 - 4.2.4.2. P2 P1'e dönüşü şu şekilde yapmalıdır "RET CHUNK"
- 4.2.5.
- 4.2.6.
- 4.2.7.
 - 4.2.7.1. Execute işlemi şu şekilde gerçekleşmektedir: P1 işlenmesi istediği resmi CHUNK'lara böler, her birini P2'ye fonksiyonu ve parametreleriyle yollar, ve bunun cevabını CHUNK'lar olarak geri alır. Bu işlemin detayları şu şekildedir: EXE isteğinden sonra, eğer P2 CHUNK'ı düzgün biçimde ele geçirmişse, EXOK mesajı döndürüp CHUNK'ı sağlam bir biçimde aldığını P1'e belirtir.

4.2.7.2. P2'nin P1'e RET gönderme işleminin detayları işe şu şekildedir: P2 CHUNK'ı işlemeyi bitirdiğinde, P1'e *RETOK* mesajı atar ve P1, P2'den CHUNK'ı RET komutuyla ister, P1 CHUNK'ı *RET CHUNK* formatında alır.

5. Özet Tabloları

5.1. Server ile alakalı komutlar

Server'a gelen	Server'ın cevabı
<i>USR XXX.XXX.XXX.XXX/PPPPP</i>	HEL
TIC	TOC
	<i>CFS XXX.XXX.XXX.XXX/PPPPP</i>
EXT	GBY

5.2. Eşler arası komutlar

Peer'e gelen	Peer'in cevabı
FUN	RFUN Fun1, Fun2, Fun3... Funn
PAR	RPAR Par1, Par2, Par3... Parn
	<i>CFS XXX.XXX.XXX.XXX/PPPPP</i>
<i>EXE myFun(Par1, Par2, Par3... Parn)</i>	<i>EXOK</i>
	<i>Sonrasında da..</i>
	<i>RET CHUNK</i>