ISDN AĞLARIN YÖNLENDİRME VE TRAFİK PLANLAMASI

Mehmet EKE

YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEMMUZ 2007
ANKARA

Mehmet EKE tarafından hazırlanan ISDN AĞLARIN YÖNLENDİRME VE TRAFİK PLANLAMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd.Doç.Dr. Nursel AKÇAM
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği ile Elektrik Elektronik Mühendisliği Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan :Yrd.Doç.Dr. Elif Uray AYDIN

Üye : Yrd.Doç.Dr. Cem NAKİBOĞLU

Üye : Yrd.Doç.Dr. Nursel AKÇAM

Üye :

Üye :

Tarih : 12/07/2007

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Mehmet EKE

ISDN AĞLARIN YÖNLENDİRME VE TRAFİK PLANLAMASI
(Yüksek Lisans Tezi)

Mehmet EKE

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Temmuz 2007

ÖZET

Teknolojinin günümüzde ulaştığı noktayla birlikte haberleşme alanında da büyük gelişmeler yaşanmış ve aynı anda çok sayıda aboneye çok farklı alanlarda hizmet verilmeye başlanmıştır. Hız ve güvenilirliğin hayati önem taşıdığı haberleşme dünyasında haberleşme ağlarının düzgün bir planlama ile oluşturulması gerekmektedir. Bu sayede kullanıcılara hizmet verilmeye başlandığında hizmetlerdeki olası kesinti miktarı en az seviyeye çekilmiş olacaktır. Hizmetlerin verilebilmesi için tahsis edilen kaynakların yetersiz planlanması işletme esnasında aksaklıklara neden olabilirken kaynakların ihtiyacın üzerinde kullanımı servis sağlayıcılar için para kaybına neden olacaktır.

Bu çalışmada, statik yönlendirmenin kullanıldığı ağlarda karmaşık bir işlem olan döngüsel çağrı kontrollü yönlendirme tablolarının oluşturulması ve oluşturulan ağ üzerindeki trafik yükünün hesaplanmasını kullanıcı için kolaylaştıran bir program geliştirilmiştir. Bu çalışmanın amacı, kaynakların yetersiz tahsisini veya aşırı kullanımını planlama aşamasında engellemektir.

Bilim Kodu : 905.1.053
Anahtar Kelimeler : Yönlendirme, ISDN ağlarda trafik yükü
Sayfa Adedi : 93
Tez Yöneticisi : Yrd.Doç.Dr. Nursel AKÇAM

ROUTING AND TRAFFIC PLANNING OF ISDN NETWORKS

(M.Sc. Thesis)

Mehmet EKE

GAZI UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

July 2007

ABSTRACT

With the level of the technology reached nowadays great developments were achieved in the communications area and many subscribers are being served simultaneously. In communications environment for which speed and reliabilty has vital importance, communication networks should be created with a proper planning. In this way when the users are being served, amount of interuptions in services will be kept on minimum level. Insufficient planning of the resources allocated for giving services may leads to some problems and overusing of the resources may cause service suppliers lost money.

In this work a program was developed. Using it, creation of loop controled routing tables that is very complicated process in networks using static routing and calculation of traffic load over the network can be achieved easily by planners. The aim of this work is to prevent insufficient allocations of resources or overusing of it at the planning stage.

Science Code : 905.1.053
Key Words : Routing, traffic load in ISDN networks
Page Number : 93
Adviser : Assist.Prof.Dr. Nursel AKÇAM

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Hocam Yrd.Doç.Dr. Nursel AKÇAM'a, ayrıca katkılarından dolayı arkadaşlarım Elektronik Mühendisi Ahmet YÜKSEL ve eşi Neriman YÜKSEL'e, mesai arkadaşlarım Yzb. Orkun DİLLİ'ye, Ütğm. Ahmet Arif ÇETİNALP'e manevi destekleriyle beni hiçbir zaman yalnız bırakmayan eşim Zahide EKE'ye teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ŞEKİLLERİN LİSTESİ x

SİMGELER VE KISALTMALAR xii

1. GİRİŞ 1

2. TEMEL BİLGİLER 4

2.1. ISDN 4

2.2. İşaretleşme 5

2.2.1. Abone santral arası işaretleşme 5

2.2.2. Santraller arası işaretleşme (ağ İşaretleşmesi) 7

2.3. SS7 Sinyalleşmesi 11

2.3.1. SS7 protokol yapısı 13

2.3.2. Mesaj transfer bölümü 14

2.3.3. İşaretleşme bağlantı kontrol bölümü 16

2.3.4. Kullanıcı ve uygulama bölümleri 17

2.3.5. İşlem yetenekleri uygulama bölümü 21

2.3.6. İşletim bakım ve yönetim bölümü 21

2.4. Yönlendirme 22

3. AĞ TOPOLOJİSİ 24

3.1. Ağ Bileşenleri 24

Sayfa

3.1.1. Sinyalleşme uç noktası SEP 24
3.1.2. Sinyalleşme geçiş noktası STP 25
3.1.3. Birleşik sinyalleşme noktası STEP 25
3.1.4. Sinyalleşme linki 25
3.2. Sinyalleşme Bağlantısı 25
3.3. Bağlantı Temelli Hizmetler 26
3.4. Bağlantısız Hizmetler 26
4. AĞ YAPISI 27
4.1. Temel Yapısal Özellikler 27
4.2. Yapıyı Belirleyen Özellikler 27
4.3. Sinyalleşme Noktalarının Numaralandırılması 28
4.4. Yönler, Yön Setleri, Yük Dağılımı ve Varış Noktaları 29
4.5. SS7 Ağları İçin Genel Yönlendirme Kuralları 30
5. UYGULAMA 31
5.1 Programın Genel Tanıtımı 31
5.2. Bağlantı Linkleri Üzerindeki Trafik Yükünün Hesaplanması 32
5.3. Kullanıcı Arayüzü 34
5.3.1. SEP butonu 35
5.3.2. STP butonu 36
5.3.3. STEP butonu 37
5.3.4. Simülasyon butonu 37
5.3.5. Link butonu 38

Sayfa

5.3.6. Seçim butonu 39

5.3.7. Yardımcı menüler 40

5.4. Programın Kullanımı 41

5.5. Programın Örnek Uygulamaları 44

5.5.1. Uygulama 1 44

5.5.2. Uygulama 2 47

5.3.3. Uygulama 3 49

6. SONUÇ VE ÖNERİLER 52

KAYNAKLAR 53

EKLER 54
EK-1 Uygulama programı için kaynak kodları 55

ÖZGEÇMİŞ 93

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. ISDN ağ 4

Şekil 2.2. Birleşik ve birleşik olmayan işaretleşme modları 11

Şekil 2.3. SS7 işaretleşme bağlantı yapısı 12

Şekil 2.4. İşaretleşme linklerinin 2Mbps hattın 16. kanalına tanımlanması 12

Şekil 2.5. SS7 sinyalleşme protokolünün katman yapısı 13

Şekil 2.6. Devre anahtarlama modunda çağrı kurulumu için Q.931 ve ISUP mesajları arasındaki ilişki 19

Şekil 2.7. Yönlendirme 22

Şekil 5.1. Düğümler arası trafik yükünün dağlımı 33

Şekil 5.2. Kullanıcı arayüzü 35

Şekil 5.3. Step butonunun kullanımı 36

Şekil 5.4. Stp butonunun kullanımı 36

Şekil 5.5. Sep butonunun kullanımı 37

Şekil 5.6. Simülasyon butonunun kullanımı 38

Şekil 5.7. Link butonunun kullanımı 38

Şekil 5.8. Seçim butonunun düğümler için kullanımı 39

Şekil 5.9. Seçim butonunun linkler için kullanımı 39

Şekil 5.10. Düğüm noktasının özelliklerinin görüntülenmesi 40

Şekil 5.11. Bağlantı linkinin özelliklerinin görüntülenmesi 41

Şekil 5.12. Yönlendirme tablolarının oluşturulması 42

Şekil 5.13. Yönlendirme tablolarının görüntülenmesi 42

Şekil 5.14. Yönlendirmelerin simülator kutusu ile gösterilmesi 43

Şekil **Sayfa**

Şekil 5.15. Bağlantı linkleri üzerindeki yük durumlarının görüntülenmesi 44

Şekil 5.16. Uygulama 1 için ağ yapısı 45

Şekil 5.17. Uygulama 1 için yük dağılım 45

Şekil 5.18. Uygulama 1 için linklerin kapasitesinin değiştirilmesi 46

Şekil 5.19. Uygulama 1 için linklerin kapasitesinin değiştirilmesi sonrası yük dağılımı 46

Şekil 5.20. Uygulama 2 için ağ yapısı 47

Şekil 5.21. Uygulama 2 için yük dağılım tablosu.. 48

Şekil 5.22. Uygulama 2 için düğüm üzerindeki trafik parametrelerinin değiştirilmesi ……. ……48

Şekil 5.23. Uygulama 2 için trafik parametrelerinin değiştirilmesi sonucu yük dağılımındaki değişiklikler ..…… 49

Şekil 5.24. Uygulama 3 için ağ yapısı ……..…. 50

Şekil 5.25. Uygulama 3 için yük dağılım tablosu……..…. 50

Şekil 5.26. Uygulama 3 için üzerinden trafik geçmeyen linklerin silinmesi…….. 51

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ITU	Uluslararası Telekomünikasyon Birliği (International Telecommunication Union)
ISDN	Tümleşik Hizmetler Sayısal Ağı (Integrated Services Digital Network)
SS7	İşaretleşme Sistemi 7 (Signaling System Num.7)
PCM 30	Darbe Kodlu Modülasyon 30 (Pulse Code Modulation 30)
TE1	Terminal Cihazı (Terminal Equipment)
LE	Yerel Santral (Local Exchange)
NT	Ağ Sonlandırıcı (Network Termination)
MF	Çoklu Frekans (Multi Frequency)
E&M	Kulak & Ağız Sinyalleşmesi (Earth&Mouth Signalink)
MTP	Mesaj transfer bölümü (Message Transfer Port)
SCCP	İşaretleşme bağlantısı Kontrol Bölümü (Signaling Connection Control Part)
OPC	Mesaj Çıkış Noktası Kodu (Origination Point Code)
DPC	Hedef Alıcı Noktası Kodu (Destination Point Code)

Kısaltmalar	Açıklama
SLS	İşaretleşme Link Seçimi (Signallink Link Selection)
TUP	Telefon Kullanıcı Bölümü (Telephone User Port)
DUP	Veri Kullanıcı Bölümü (Data User Port)
CIC	Devre Tanımlama Kodu (Circuit İdentification Code)
CRV	Arama Referans Numarası (Call Referans Value)
IAM	Başlangıç Adres Mesajı (Initial Adress Message)
OMAP	İşletim Bakım ve Yönetim (Operations Maintenance and Administration Part)
SP	Sinyalleşme Noktası (Signaling Point)
STP	Sinyalleşme Aktarma Noktası (Signalling Transfer Point)
SEP	Sinyalleşme uç noktası (Signalink End Point)
STEP	Birleşik Sinyalleşme Noktası (Signalink Transfer and End Point)
SL	Sinyalleşme Linki (Signalink Link)
PBX/ PABX	Özel Kurum Santralı (Private Branch Exchange)
CN	Çağrı Sayısı (Call Number)
CD	Çağrı Süresi (Call Duration)
MP	Ölçüm Dönemi (Measurament Periyod)
ICR	Düğüme gelen çağrı oranı (Incoming Call Rate)

1. GİRİŞ

İnsanoğlunun sürekli artan ihtiyaçları her alanda olduğu gibi haberleşme alanında da teknolojiyi gelişime zorlamaktadır. Artık insanlar için uzaklardaki bir yakınının yalnızca sesini duymak yeterli olmamaktadır. Yakın bir zamanda haberleşme ağındaki her kullanıcı için görüntülü iletişim sıradan bir talep haline gelecektir. Dolayısıyla her kullanıcı için daha fazla bant genişliği ve kaynak tahsisi yapılması gerekecektir. Günümüz iş çevrelerindeki rekabet ortamı servis alınan ağdan kesintisiz ve hızlı hizmetler beklemektedir. Bugünün haberleşme hizmetlerini yönlendiren büyük servis sağlayıcı şirketler artık sunulan hizmetlerde bütünleşmeye gitme eğilimindedirler. Yakın bir zamanda kullanıcılar IP TV uygulamaları ile tek bir cihaz üzerinden televizyon seyredebilecek, kendi kanalını yaratabilecek, servis sağlayıcının oluşturacağı bir veri tabanında kendi film kolleksiyonunu saklayabilecek, internetin tüm olanaklarını kullanabilecek hale gelecektir.

Bu hizmetlerin sunulabilmesi için servis sağlayıcıların öncelikle bir heberleşme ağı oluşturması gerekmektedir. Oluşturulan ağda, sunulan hizmetler tüm abonelere kesintisiz ve hızlı bir şekilde ulaştırılmalıdır. Bu hizmetlerin neler olduğuna ‘Temel Bilgiler’ kısmında, ISDN başlığı altında değinilmiştir. Ağın omurgasını oluştaracak olan geçiş düğüm noktalarının ve bu düğüm noktalarının kullanacakları anahtarlama yöntemleri, ağın kullanıcılara sunacağı hizmetler ve maliyet göz önünde bulundurularak belirlenir. Bu anahtarlama teknikleri yönlendirme olarak isimlendirilir. Yönlendirme konusu bu döküman içinde aynı başlık altında anlatılmıştır. Haberleşme ağlarında genel olarak dinamik yönlendirme ve durağan yönlendirme olmak üzere iki tür yönlendirme tekniği kullanılmaktadır.

Dinamik yönlendirme çoğunlukla internet hizmeti veren ve IP, ATM veya halen çalışma aşamasında olan MPLS teknolojisi ile çalışan ağlarda kullanılmaktadır. Bu tip ağların omurgasını yönlendiriciler (router) oluşturmaktadır. Her yönlendirici, kendisine gelen veriyi, hedefe doğru hangi

yoldan göndereceğine kendisi karar verir. Yani belirli bir hedefe doğru iletilen verinin takip edeceği yol, ağın durumuna göre değişir buna yönlendiricinin kendisi karar verir. Bu işlemlerin yapılması için omurgayı oluşturan tüm yönlendiriciler birbileri ile etkileşim halinde olmalıdır. Yani aralarında bir işaretleşme olmak zorundadır. İşaretleşme belirli bir protokol dahilinde gerçekleştirilmelidir. İşaretleşme protokolleri, bir nevi, cihazların kendi aralarındaki konuşma dilidir. Bu alandaki teknolojinin ilerlemesiyle birlikte farklı protokoller geliştirilmiştir. Bu protokollerden biri olan RIP (Routing Internet Protocol), verinin hedefe doğru ilerleyişi esnasında en az düğümden geçmesi, yani en az atlama ile hedefe ulaşması mantığı üzerine kuruludur [1]. Bu çalışmanın uygulama kısmında geliştirilen simülasyon programı da bu protokol mantığı ile çalışmaktadır.

Durağan yönlendirme ise çoğunlukla ses ve görüntü hizmeti verilen haberleşme ağlarının omurgasını oluşturan ve SS7 (Signalink System 7) işaretleşme protokolleri ile çalışan düğüm santrallerinde kullanılmaktadır. SS7 işaretleşmesi ile ilgili detaylı bilgi ‘Temel Bilgiler’ kısmında verilmiştir. Durağan yönlendirme ile çalışan düğüm santrallerinde yönlendirme bilgisi bir operatör tarafından önceden tanımlanmalıdır. Yani cihazlar yön bilgisine kendileri karar veremezler. Başka bir ifade ile eğer operatör belirli bir hedef için yön bilgisini uzun atlamalar olacak şekilde girerse daha kısa bir yol olsa bile veri operatörün tanımladığı yolu izleyecektir. Bir başka konu ise omurgayı oluşturan düğüm santrallerinin birbirlerine bağlantısını sağlayan linkler ve bu linkler üzerinde oluşacak trafik yüküdür. Bu linkler ve kapasiteleri trafik yükünü kaldıracak şekilde planlanmalıdır [2].

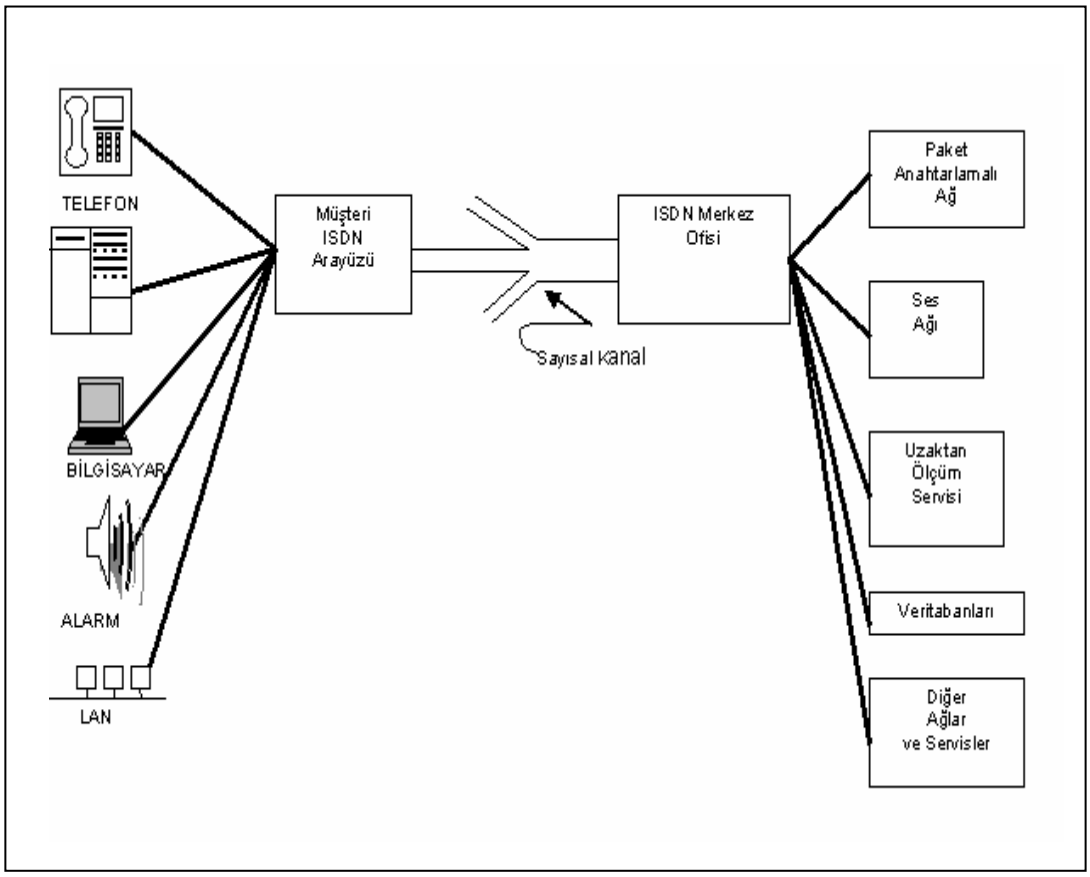
Bu çalışmada durağan yönlendirme kullanılan geçiş santrallerinin oluşturduğu bir ağı simüle eden bir program geliştirilmiştir. Kullanılacak düğüm santralleri ve bağlantı linkleri program üzerinde oluşturulabilmektedir. Program RIP protokol mantığını kullanarak her düğüm için yönlendirme tablolarını oluşturmaktadır. Aynı zamanda bağlantı linkleri üzerindeki trafik yüklerinin dağılımını hesaplayarak kullanıcıya gerekli kapasite bilgisini

vermektedir. Çalışmanın amacı, dinamik yönlendirme mantığı kullanılarak, operatörler için karmaşık ve zor bir işlem olan durağan yönlendirme tablolarının, bir kullanıcı arayüzü ile kolay ve hızlı bir şekilde oluşturulması, aynı zamanda yine operatörler için uzun zaman gerektiren, trafik yükünün linkler üzerindeki dağılımının hesaplanmasını kolaylaştırmaktır.

2. TEMEL BİLGİLER

2.1. ISDN

ISDN; ses, metin, görüntü ya da veri gibi birden fazla servisin iletilmesini sağlayan sayısal ağdır (Şekil 2.1).



Şekil 2.1. ISDN ağda verilen hizmetler [3].

En basit şekliyle ISDN, aynı çiftli telefon kablosu üzerinden hem ses hem de verinin taşınmasına müsaade eden yerel telefon hattının geliştirilmiş halidir. Başka bir yaklaşımla ISDN geniş bir yelpazede veri ve haberleşme hizmetleri sağlayan bir ağ olarak ifade edilir. Bu ağ tamamen sayısaldır ve ağdaki tüm cihazlar ve uygulamalar sayısal olarak temsil edilirler. Yani telefon, PC, televizyon, santral gibi cihazlardan gelen bilgiler ağ santrali

tarafından bit dizileri olarak algılanırlar. Dolayısıyla bilginin tümü aynı ağ teçhizatı tarafından anahtarlanabilir ve taşınabilir. Daha gerçekçi bir bakışla ISDN, hem ağ hizmetlerine erişimi sağlayan abone hattının teknolojik değişimi, hem de ağın bizzat kendi altyapısını tarif eder. Bu yaklaşımda ISDN abonelerinin ISDN özelliği olmayan abonelerle uyum içinde çalışabildiği, ağın sunduğu her türlü hizmetin ISDN özellikli veya ISDN özellikli olmayan tüm kullanıcılar tarafından paylaşılabilir [3].

2.2. İşaretleşme

İşaretleşme; iletişim ağlarında bir kullanıcının diğer bir kullanıcıya bağlanması için gerekli bilgilerin belirlenmiş protokollere göre alınıp gönderilmesidir .

Kullanıcının servis isteği, konuşma yolunun kontrolü, adres bilgisi, ücretlendirme ve konuşmanın çözülmesi gibi işlevsel görevler işaretleşme sayesinde gerçekleştirilir .

Bir ISDN ağda uçtan uca iki kullanıcı arasında teknik anlamda birbirinden farklı fakat birbirine uyumlu işaretleşmeler kullanılır. İşaretleşmeler fonksiyonel olarak ikiye ayrılır [3,4].

2.2.1. Abone santral arası işaretleşme

Terminal cihazı ile santral arasındaki haberleşmeyi sağlar. Ahizenin kaldırılması, santralden çevir sesi alınması gibi bilgiler bu işaretleşme kapsamındadır.

Kullanıcı-ağ terimi bize, bu işlevlerin bir ISDN kullanıcı Terminal Cihazı (TE, Terminal Equipment) ile Yerel Santral (LE, Local Exchange) arasında bir ara yüz üzerinde gerçekleştiğini ifade eder. Bir cihazın ISDN kullanıcı terminali olarak değerlendirilebilmesi için bu konu içinde anlatılan işlevleri yerine

getirebiliyor olması gerekir. Bununla birlikte, kullanıcı terminali bir Özel Kurum Santrali (PBX, Private Branch Exchange) veya bir Ağ Sonlandırıcı (NT, Network Termination) olabilir. Bahsedilen işlevleri yerine getiremeyen cihazlar (analog telefon, eşzamansız yapıda veri terminalleri, modem vb.) da bir Terminal Dönüştürücü (TA, Terminal Adaptor) ile bağlanmış olabilir. Bu durumda TA, ISDN kullanıcı terminal durumunda olur.

Dikkat edilmesi gereken bir nokta da kullanıcı-ağ ara yüz işlevlerinin sadece bir terminalle ağı temsil eden santral arasında olduğudur. Bu işlevler ve onlarla ilişkili protokoller santraller arasında kullanılanlardan farklıdır ve ağ içerisinde ilerleyemezler. Santraller arasında kullanılan protokoller SS7 işaretleşmesi kapsamındadır. Kullanıcıdan ağa gönderilen bir mesajın ağ içerisinde başka bir yere iletilebilmesi için önce ISDN protokol tarafından okunup SS7 protokolüne çevrilmesi gerekir.

Temel çağrı kontrolü, bir çağrının basit olarak kurulup, devam ettirilip, sonlandırılacağı anlamına gelir. Bir çağrı esnasında iki kullanıcı arasında, ağ üzerinde bir yol yaratılır.Temel çağrı kontrolünde, arama direkt olarak arayan taraftan aranan tarafa doğru kurulur. ITU-T' nin temel çağrı kontrolü için protokolleri Q.931 tavsiyeleri içinde yer alır .

Analog telefonlarda kullanıcı arama isteğinde bulunduğunu ahizeyi kaldırarak bildirir. Bu hat üzerinde bir elektriksel kısa devre oluşturacak ve Yerel Santral (LE, Local Exchange)' den gelip kullanıcı telefonu üzerinden akarak yeniden LE ye dönen bir akım oluşmasına neden olacaktır. Bu döngü sonucunda ağ kullanıcının bir arama talebinde bulunduğunu anlar. Bu bir çağrı başlangıcı sinyalidir. Sonra ağ kullanıcıya hedef adresinin belirlenmesi için gerekli olan daha fazla sinyali almaya hazır olduğunu gösteren bir çevir sesi gönderir [3,4].

ISDN işaretleşme protokolleri gerekli işaretleşme bilgisiyle kodlanmış sayısal mesajların kullanıcı ile ağ arasında alınıp verilmesine dayanır. Artık sayısal

ağda bildiğimiz ses ve ton sinyallerine ihtiyaç kalmayacaktır.

2.2.2. Santraller arası işaretleşme (ağ işaretleşmesi)

Haberleşme ağ işaretleşme sistemleri, kullanıcıların haberi olmadan, ağ santralleri için yönlendirme, link durumu ve bağlantı kontrolü bilgilerinin alınıp verilmesi mekanizmasını sağlayan en önemli bileşendir. Devre anahtarlamalı bir ağdaki uçtan uca bağlantı bir seri santraller arası hattı da kapsar. Bu bağlantıların nasıl sağlandığı zaman içerisinde geliştirilen işaretleşme tekniklerine göre değişmektedir.

Ağ işaretleşmesi genel olarak 3 büyük fonksiyonu gerçekleştirir:

Gözetim: Devrenin durumunu izler, telefon ahizesinin kaldırılıp kaldırılmadığına ve telefon hattının kullanıma açık olup olmadığına bakar.

Adresleme: Yönlendirme bilgisini sağlar; aranan abonenin numarasına göre yönlendirme yapar.

Arama Bilgisi: Aramanın geçtiği evrelerin durumunu; telefon çalıyor sesi, meşgul tonu işaretleri olarak belirtir .

Ağ işaretleşmesini evrimsel ve tür olarak üç gruba ayırmak mümkündür.

Bant içi işaretleşme (İn Band Signaling)
Bant dışı işaretleşme (Out of Band Signaling)
Ortak kanal işaretleşmesi (Common Chanel Signaling)

Bant içi ve bant dışı işaretleşme kanala bağlı işaretleşme olarak da bilinir.

Bant içi işaretleşme

1970'li yıllardan önce bir telefon görüşmesi ağ boyunca bu işaretleşme türü kullanılıyordu. Ağ sinyalleri 300-3400 Hz.' lik ses bandında taşınıyordu.

Ağ işaretleşmesinde santraller arasında yukarıdaki fonksiyonlar için değişik tonlar kullanılır.Tek frekans (SF, Single Frequency) ve çoklu frekans (MF, Multi Frequency) tonları bunun örnekleridir. Eskiden 2600 Hz.'lik tek frekans tonu, hattın boş durumda olduğunu belirtiyordu, eğer bu ton yoksa hat kullanımda demekti. MF tonları ise genellikle adres bilgilerini taşımak için kullanılıyordu.

Bu işaretleşme bir görüşme için gerekli fiziksel yol boyunca yol üzerindeki tüm hatların açık olmasını gerektirmektedir. Eğer son hat doluysa bu bilginin ağ tarafından öğrenilmesinden önce, tüm diğer yollar kurulduğu için boşuna hatlar meşgul edilmiş olacaktır. Bu işaretleşme teknikleri dar bir sahada işlevsellik sağlar. Çünkü sınırlı sayıda kod setleri (tipik olarak 16 ton) kullanımı vardır. Bağlantının kurulma süreleri 10 ile15 saniye'yi bulmaktadır. Ayrıca kötüye kullanıma açık sistemlerdir. Kullanıcılar ödemeli bir telefon makinesinin içine bir jeton atıldığını zannettirecek tonlar üreten değişik cihazlar yapmışlardır.

Bant dışı işaretleşme

İletim sistemlerinin ortaya koyduğu kısıtlamalardan veya bedavacılardan kaynaklanan ekonomik konular, telefon endüstrisinde bant dışı işaretleşmesini ortaya çıkardı. Basit bir şekilde bant dışı işaretleşmede işaret mesajları kullanıcı ses bandı dışından gönderilir.

İlk bant dışı işaretleşme sistemleri aynı kablo çiftlerini (E&M işaretleşmesi olarak adlandırılır) veya 300-3400 Hz ses bandı dışında (3700 Hz işaretleşmesi) frekans kanalını kullanır. Bu sistemler konuşma kanalı başına

bir işaretleşme devresi gerektirir ve işaretleşmenin geneli bant içi işaretleşmedekine benzer. Veri iletimindeki gelişmeler atanmış bir veri kanalının işaretleşme için kullanımını (fiziksel olarak ayrı bir devreye ya da ton devresinden ziyade) beraberinde getirdi.

Ortak kanal işaretleşmesi

Ortak kanal işaretleşmesi, bir tür bant dışı işaretleşme olup, kullanıcı ses kanallarından ayrı işaretleşme kanalları kullanılarak işlemci donanımlı santral merkezleri arasında işaretleşme bilgilerini taşımak için tasarlanmıştır. Bu işaretleşme tekniği ağın sunduğu hizmetlerin hızlı bir şekilde tahsis edilmesine, test edilmesine ve çözülmesine imkan tanır. Ortak kanal işaretleşmesinde yolun ulaşılabilir olup olmadığının belirlenmesi için ağda bir arama yolunun her bölümünü kontrol edebilir. Eğer yol açıksa gerekli kaynaklar hemen tahsis edilerek görüşme sağlanabilir. Bu çok hızlı bir şekilde bağlantının kurulmasına, geri dönüşlerdeki zaman kaybının minimize edilmesine ve daha fazla etkin yol tanımlamaya müsaade eder. Ortak kanal işaretleşmesinin kullanımı ile ortalama bağlantı kurulma süresi 15 ile 20 saniye'den 3 ile 5 saniye'ye düşmüştür.

Ortak kanal işaretleşme ağlarından SS7 işaretleşme sistemi ITU-T tarafından uluslar arası düzeyde tanımlanmış, genel amaçlı ortak kanal işaretleşme sistemidir. Ulusal ve uluslararası ağlarda kullanılabilecek sayısal ağlara ve devamında da ISDN ve yeni uygulamalara (GSM, TMN, IN, Multimedya vs…) alt yapı teşkil edecek bir sistemdir. Ortak Kanal işaretleşmesi kullanıcı ve ağ sağlayıcıya bir çok faydalar sunmaktadır:

İşaretleşme bant dışı olduğu için ve birden fazla hat'ın işaretleşmesi tek bir işaretleşme kanalı üzeriden çoğullandığı için uzun mesafede bant genişliği koruması sağlar.

Maliyetler azalır. Çünkü ihtiyaç duyulan donanım azdır. Bant içi işaretleşmede, her bir kullanıcı devresi için ayrı birer işaretleşme hizmeti gerekmektedir.

- İşaretleşme mesaj paketleri ile yapılır.
- Konuşma süresince işaret iletimi mümkündür.
- Etkin ağ yönetimi ve bakım yapılabilmektedir.
- Etkin yönlendirme tabloları ve işaretleşme linki yedeklemesi yapılabilmektedir.

Ortak kanal işaretleşmesinin bileşenleri

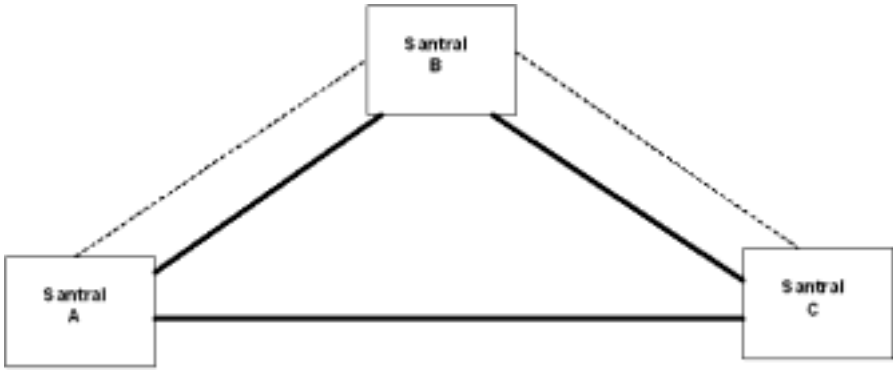
Bu tür ağların bileşenlerinde; işaretleşme linkleri ile birbirlerine irtibatlı İşaretleşme Düğüm Noktaları (Nodes) mevcuttur. Bu işaretleşme düğüm noktaları işlemci kontrollü santrallerdir ve bu santraller arasında gözlem, adres, çağrı ilerleme ve diğer ağ sinyalleri alınıp verilir. Bu işlemi yapan her bir santrale Servis Anahtarlama Noktası da denir. Ayrıca üzerine aldığı sinyalleri paketler halinde diğer sinyal noktalarına gönderen Sinyal Transfer Noktaları ve çeşitli veri tabanlarına bağlı Servis Kontrol Noktaları gibi bileşenler de vardır.

Ortak kanal işaretleşmesi modları

Bir işaretleşme ağında işaretleşme modu kavramı, işaretleşme mesajlarının gerçek yolu ile işaretlerin ait olduğu bilgi akış yolu arasındaki ilişkiyi belirtir. Ortak kanal işaretleşmesinde genelde iki işaretleşme modu vardır.

Birleşik işaretleşme Modu (Associated Signalling Mod): İki sinyal noktası (santral) arasındaki bilgi akışına ait işaretleşme mesajları iki santrali birbirine doğrudan bağlayan hat üzerinden taşınır. Kullanıcı ile ağ arasındaki arayüzde D-kanalının kullanımı birleştirilmiş işaretleşme moduna örnektir.

Birleşik olmayan işaretleşme modu (Nonassociated Signalling Mod): burada iki santral arasındaki kullanıcı bilgilerinin aktığı hatlarla işaretleşme linkinin birbiri ile aynı fiziksel hattı takip etmesi gerekmez. Şekil 2.2'de görüldüğü üzere A'dan C'ye direkt değil de endirekt olarak B santralı üzerinden işaretleşme linki tanımlanabilir.



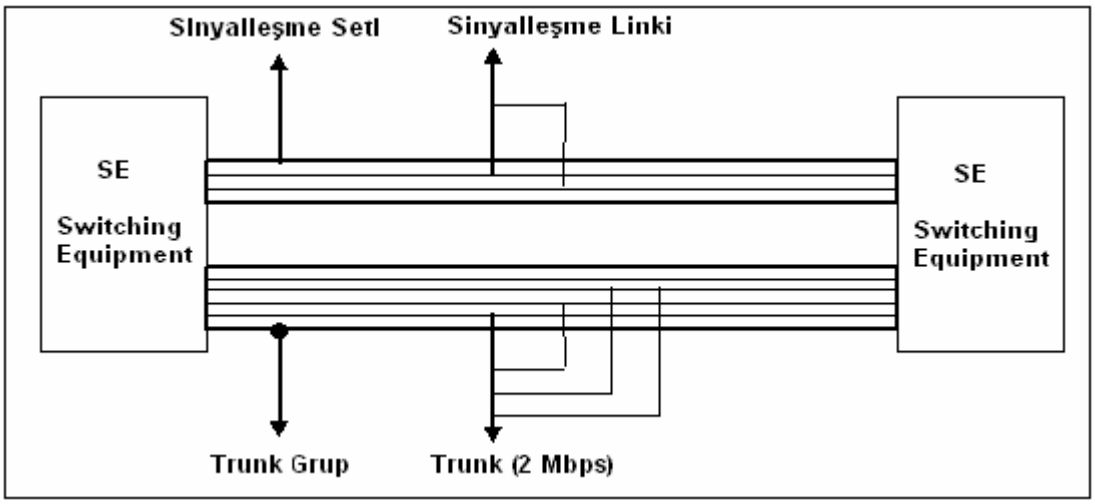
Şekil 2.2. Birleşik ve birleşik olmayan işaretleşme modları

2.3. SS7 İşaretleşmesi

ITU-T Q.700 Tavsiyesi Ortak Kanal İşaretleşmesinin ve SS7 İşaretleşmesinin bir özetini yapar. Kalan Q.700 serileri ise SS7 protokol ve işlevlerini içerir .

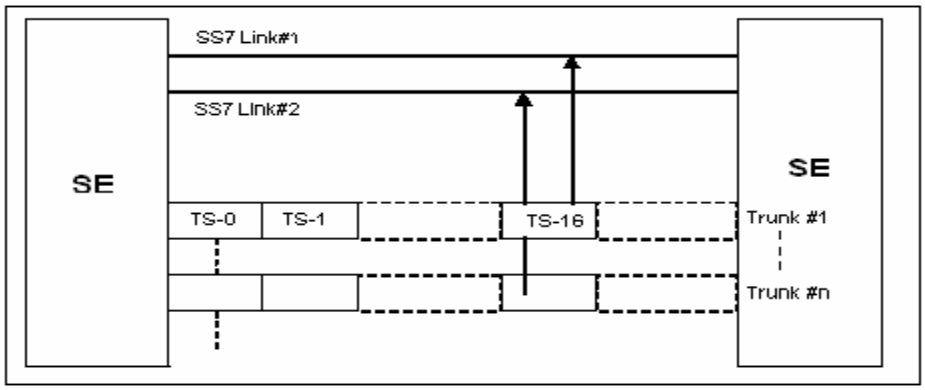
SS7 protokolleri katmanlı bir yapıya sahiptir. Gelecekteki gelişmeler için esnek bir yapısı vardır. Bu katmanlı yapı herhangi bir katmanda yapılacak değişikliğin diğer katmanları etkilemeyeceği anlamına gelir. Ayrıca işaretleşme ağına yeni üst seviye servis ve uygulamalar ilave etme imkanı vermektedir.

SS7 birleşik ve yarı-birleşik işaretleşme modlarını kullanacak şekilde tasarlanmıştır. Çünkü sıralı olmayan mesajları sıraya sokacak ilave prosedürler yoktur. Birleşik ve birleşik olmayan işaretleşme türleri mesajların sıralı olarak yerlerine varmalarını sağlar [5].



Şekil 2.3. SS7 işaretleşme bağlantı yapısı

Şekil 2.3'de, iki santral arasına tanımlı işaretleşme linkleri ve hatlar gösterilmiştir. Hatlar santraller arası konuşma kanalları olarak tanımlanır. 2Mbps çerçeve yapısı, 64 Kbps lik 32 adet kanaldan oluşur. Her bir kanal tek bir telefon görüşmesi için ayrılır. 2 Mbps çerçevede 16. kanal işaretleşme için

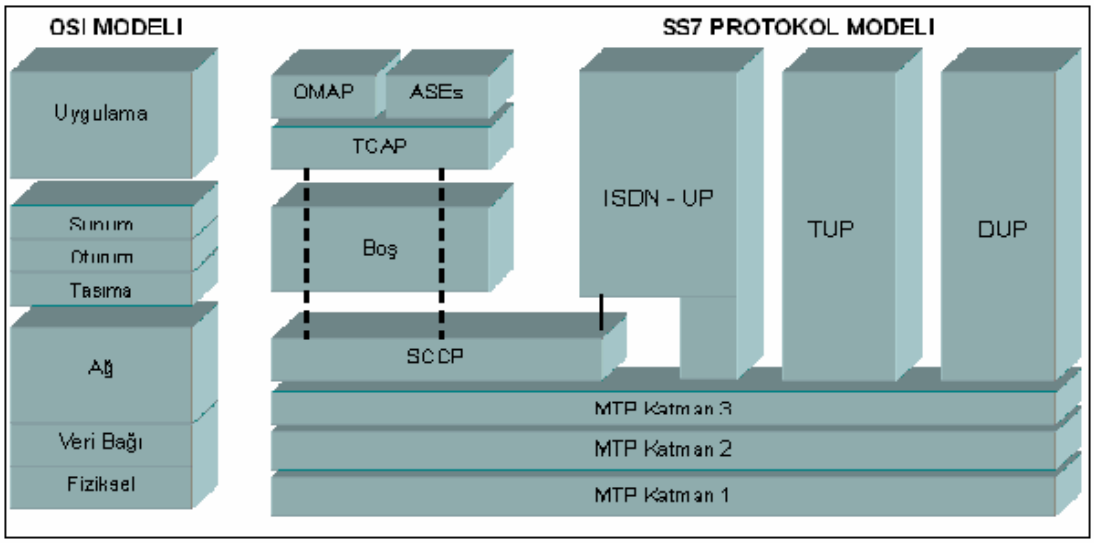


Şekil 2.4. İşaretleşme linklerinin 2Mbps hattın 16. kanalına tanımlanması.

ayrılır. İşaretleşme linki mesaj alış verişi esnasında bu kanalı kullanır. Şekil 2.4 SS7 işaretleşmesinde bir hat grubu içinde tanımlı n adet hattan 1 tanesinin 16. kanal işaretleşme için ayrıldığında, o hat grup içindeki diğer hatlar tanımlanan bu işaretleşme linkini kullanabilirler [6].

2.3.1. SS7 protokol yapısı

Şekil 2.5'de görülen SS7 protokol yapısı içinde Mesaj transfer bölümü (MTP, Message Transfer Port) daha çok OSI'nin düşük ilk üç katmanındaki protokolleri içerir. Bunlar fiziksel, veri bağlı ve bağlantısız ağ katman fonksiyonlarıdır. MTP katmanı işaretleşme mesajlarının güvenilir bir şekilde transfer ve teslimini sağlar .



Şekil 2.5. SS7 işaretleşme protokolünün katman yapısı [6].

İşaretleşme bağlantısı Kontrol Bölümü (SCCP, Signaling Connection Control Part) OSI adresleme yetenekleri ve bağlantıya yönelik bilgi transferini içeren bazı ağ katmanı protokol fonksiyonlarını sağlar .

Kullanıcı ve Uygulama Bölümleri (User and Applications Part) ISDN'deki ses ve ses olmayan anahtarlamalı servisler için uçtan uca işaretleşme fonksiyonu sağlar. Bazı farklı kullanıcı veya uygulama bölümleri tek bir MTP veya SCCP protokol uygulaması üzerinde çalışabilir [6].

2.3.2. Mesaj transfer bölümü

MTP, OSI'nin ilk üç zincir katmanına denk gelen protokol katmanlarına sahiptir (Şekil 2.5). SCCP ve diğer kullanıcı uygulama bölümlerine işaretleşme mesajlarının güvenli bir şekilde iletim ve teslimini garanti eden bağlantısız servis sağlar. MTP ağ içindeki link ve santral arızalarını tespit eden mekanizmaya da sahiptir .

MTP katman 1 protokolü

İşaretleşme Veri Hattı (Signalling Data Link) olarak adlandırılır. İşaretleşme sistemi tam-çiftli sayısal hat üzerinde ve 64 Kbps hızda çalışacak şekilde tasarlanmıştır. Bazı kullanıcı bölümü (User Part) uygulamalarında kabul edilemez uzun süreli gecikmeler olmasına rağmen eski anahtarlama ve iletim donanımı ve eski işaretleşme sistemleri ile uyum maksadıyla telefon çağrı kontrolu için 4.8 Kbps.'lık hıza kadar çalışabilmektedir. Ayrıca 1.544 Mbps. ve 2.048 Mbps.lik hızlarda da çalışmaktadır .

MTP katman 2 protokolü

İşaretleşme Hattı (Signalling Link) olarak adlandırılır ve OSI'nin ikinci katmanı olan Veri Bağı katmanına karşılık gelir. MTP Katman 2 bite yönelik protokol kullanır. Diğer bite yönelik protokollerde olduğu gibi Sinyal Birimi olarak adlandırılan MTP Katman 2 çerçeveleri başlangıcında altı tane 1 ve iki tane 0 dan oluşan bir işaretçi vardır. MTP sinyal birimleri içerisinde adres bölümü yoktur. Çünkü haberleşme sadece noktadan noktaya linkler üzerinden yapılmaktadır .

MTP katman 3 protokolü

İşaretleşme Ağı fonksiyonlarını tanımlar ve OSI'nin Ağ Katmanının alt yarısına denk gelir. Ağın işaretleşme noktaları arasında mesajların

iletiminden sorumludur. Bu katmanda yapılan iki genel görev vardır; mesaj kotarma ve ağ yönetimi. Mesaj kotarma işlemi mesajın içerisinde yer alan Mesaj Çıkış Noktası Kodu (OPC, Origination Point Code), Hedef Alıcı Noktası Kodu (DPC, Destination Point Code) ve İşaretleşme Link Seçimi (SLS, Signaling Link Selection) değerlerine göre yapılır. SLS, SS7 İşaretleşmesinde yük paylaşımının yapılmasını sağlayan bölümdür. Herhangi belirli bir hedef noktaya tanımlanan yön yükü dengelemek için birden fazla işaretleşme linki içerisinden herhangi birini kullanabilir. İşaretleşme ağının mesaj fonksiyonları şunlardır:

Mesaj Ayrımı: MTP Katman 2'den alınan mesajın DPC'sine bakar ve mesajın kendisine mi yoksa başka bir sinyal noktasına mı ait olduğunu kontrol eder. Kendisine aitse mesaj dağıtım fonksiyonuna, değilse yönlendirme fonksiyonuna aktarır.

Mesaj Dağıtımı: MTP Katman 2'den alınan mesaj bu sinyal noktasına aitse SIO alanına bakarak onu uygun MTP kullanıcısı veya MTP Katman 3 fonksiyonuna aktarır.

Mesaj Yönlendirme: Eğer MTP Katman 2'den gelen mesaj başka bir sinyal noktasına aitse veya mesaj kendi sinyal noktasından çıkarsa onu DPC ve SLS bilgisine göre aktarır .

MTP Katman 3 fonksiyonu aynı zamanda ağ yönetimi işlevlerini de yerine getirmektedir. Bu fonksiyonların maksadı işaretleşme linki veya işaretleşme noktası arızalarında işaretleşme ağının yeniden konfigürasyonunun sağlanabilmesi ve bir tıkanıklık durumunda trafiğin kontrol altına alınabilmesidir. Bu tür durumlara karşı tepki olarak yapılacak ağın yeniden konfigürasyonu kayıplara sebebiyet vermeden, dublikasyonlar, sıra dışı mesajlar ve aşırı gecikmeler yaratmadan tamamlanmalıdır. İşaretleşme ağ yönetimi fonksiyonları şunlardır:

İşaretleşme Trafik Yönetimi: İşaretleşme trafiği link arızası veya tıkanıklık durumunda alternatif yönlere verilir; bu esnada kayıp veya dublikasyonların olmaması sağlanır.

İşaretleşme Yön Denetimi: İşaretleşme ağının durumu hakkında diğer işaretleşme noktalarına işaretleşme yönlerini kapatmaları veya açmaları için bilgi dağıtır.

İşaretleşme Hat Yönetimi: Arızalanan linkleri yeniden başlatır, yeni linkler aktif eder ve hatalı linkleri işletimden çıkarır .

İşaretleşme ağının fonksiyonel açıklamalarının büyük bir bölümü hat arızaları ve tıkanıklık durumlarının üstesinden gelmek için uygulanan prosedürler ile ilgilidir. Prosedürler bir link koptuğu zaman anında hattın servisten silinmesi , trafiğin başka bir yöne aktarılması ve onarımdan sonra linkin tekrar aktif edilmesi için belirlenmiştir. Ağ güvenirliği için önemli bir kaygı vardır; yılda 10 dk.dan daha uzun süreli hiçbir linkin arızalı kalmaması ve yüzde olarak da %99.998 'lik bir kullanılabilirlik oranı olmalıdır. SS7 işaretleşmesinde bu hedefe donanım yedeklemesi, dinamik olarak yeniden ağ konfigürasyonu ve yeniden yönlendirme yetenekleri kullanılarak ulaşılmıştır .

2.3.3. İşaretleşme bağlantı kontrol bölümü

MTP başlangıçta telefon ağ işaretleşmesinin gerçek zamanda ihtiyaçlarının karşılanması için tasarlanmıştır (Şekil 2.5). Bu nedenle, bağlantısız bir ağ servisi sağlamar. Bazı uygulamalar MTP'nin sağladığından daha büyük ve bağlantıya yönelik iletim yeteneği gerektirmektedir. MTP'yi tamamlayan ve OSI'nin 3 ncü seviyedeki ağ katmanına tam denk olmasını sağlayan bölüm SCCP bölümüdür. MTP ve SCCP ikisi birlikte SS7'nin ağ servisleri bölümünü oluştururlar .

SCCP'nin sağladığı en büyük fayda geliştirilmiş adresleme fonksoniyelliğidir. MTP mesaj adresleme yeteneği bir mesajın bir düğüme DPC ve OPC'ler kullanılarak teslimini sağlar ve düğümde bir MSU'nun SIO alanındaki 4 bitlik bir belirtici kullanan sınırlı dağıtım yeteneği vardır. Bu yetenek çok düşük seviyedeki servisler için yeterli olmasına rağmen, daha fazla uygulama kullanıldığında ve özellikle SS7 için tanımlanmış kullanıcı bölümleri için yeterli olmadığı görülmüştür. Uygulamaların sayısı arttıkça MTP'nin sınırlı kapasiteli adres bölümü sadece mesajın bir düğüme teslim edilebilmesine yetecek fakat bu mesajla birleştirilmiş uygulamanın belirtilmesine yetmeyecektir .

Bu problemi aşmak için MTP adresi üzerine bir de Subsystem Number (SSN) ilave bilgisi konur. Bu bilgi her düğümde bulunan bir SCCP kullanıcısının dahili numarasıdır. OPC+SSN arayan tarafın adresini oluşturur. DPC+SSN de aranan tarafın adresini oluşturur .

Diğer bir SCCP gelişimi ise adres olarak genel başlıklar kullanılabilmesidir. Bu adres 800 gibi bir numara olabilir ve MTP yönlendirmesi için kullanılamaz. Ancak SCCP bunu çevirme işlemi ile DPC+SSN numarası şekline çevirir ve ilgili santrala iletir, hedef santral da bu mesaj lokal SCCP'ye aktarılır ve o da doğru ağ servis kullanıcısına teslim eder.

2.3.4. Kullanıcı ve uygulama bölümleri

SS7'nin kullanıcı ve uygulama bölümleri OSI'nin uçtan uca yüksek katmanlarına karşılık gelir (Şekil 2.5). Kullanıcı ve uygulama bölümlerinin her biri bir bütün ve birbirinden bağımsızdır. Kullanıcılar uçtan uca işaretleşme hizmeti sağlayan MTP veya SCCP ile direkt olarak haberleşirler. Bazı kullanıcı ve/veya uygulama bölümleri paralel olarak tek bir MTP veya SCCP bağlantısı üzerinde çalışabilir .

Orjinal olarak SS7 için belirlenen iki kullanıcı bölümü Telefon Kullanıcı Bölümü (TUP, Telephone User Port) ve Veri Kullanıcı Bölümü (DUP, Data User Port) ‘dür. TUP, normal yurtiçi ve uluslararası telefon haberleşmesinin kontrolü için gerekli işaretleşmeyi belirtir. DUP devre anahtarlamalı veri ağları için tasarlanmıştır .

ISDN kullanıcı bölümü

ISDN kullanıcı bölümü (ISUP veya ISDN UP) ISDN devre anahtarlamalı taşıyıcı servisleri için gerekli temel işaretleşmeyi sağlar. ISUP,TUP’un sağladığı sese yönelik işaretleşme servisine ilave olarak ses olmayan aramalar ve uçtan uca belirtilmiş ISDN ilave servisleri için de fonksiyonlar sağlar.

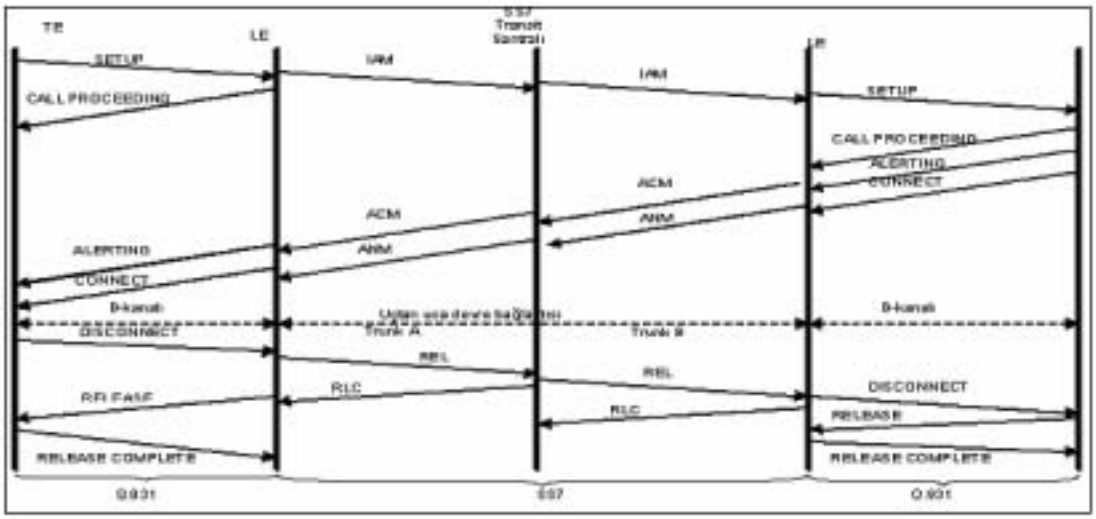
ISUP, MTP veya SCCP tarafından sağlanmış olan iletim servislerini de kullanabilir. MTP servisleri santraller arasındaki aramayla ilgili işaretleşme mesajlarının aktarımında, SCCP ise uçtan uca işaretleşme gibi ilave bağlantı servisleri için kullanılır .

SS7 ağ içinde bir ağdır. Uç kullanıcılar ISDN servislere kendi lokal santralleri ile ulaşır. Bu lokal santraller de SS7’nin ISUP servislerini kullanarak bu servisleri tesis ederler. ISUP, ISDN uç kullanıcılar için şeffaftır yani ISUP onlar için geçirgendir. Aynı santral içinde başlayan ve biten aramalarda ISUP kullanılmaz. ISUP mesajları işaretleşme linkinde sinyal unitleri (birim) olarak taşınır.

MTP ve SCCP’nin servislerini kullanan ISUP, iki ISDN lokal santral arasında mantıksal bağlantılar sağlar. Bu bağlantılar bir Devre Tanımlama Kodu (CIC, Circuit İdentification Code) ile tanımlanırlar. Bir ISDN aramanın kurulması ve korunması için bazı adresler gereklidir; ISDN terminal tarafından çevrilen telefon numarası SCCP ve/veya MTP tarafından bağlantı kurulması esnasında sadece yönlendirme için kullanılır. CIC, ISUP tarafından SS7

bağlantısına ilişkin olarak kullanılır ve ISDN lokal santrali ile ISDN terminali arasındaki bağlantının tanımlanabilmesi için bir Arama Referans Numarası (CRV, Call Referans Value) kullanılır.

Şekil 2.6 basit bir ISDN devre anahtarlamalı aramanın kurulması için örnek mesaj alışverişini göstermektedir. Bu diagram Q.931 ile ISUP mesajları arasındaki ilişkiyi de göstermektedir.



Şekil 2.6. Q.931 ve ISUP mesajları arasındaki ilişki [6].

Bir kullanıcı ISDN arama başlattığında (telefon ahizesini kaldırarak) arayan TE ISDN D-kanalı üzerinden bir Q.931 SETUP mesajını lokal santraline gönderir. Arayan abonenin lokal santrali SS7 ağında bulunan ve bir sonraki santrale doğru tanımlanmış bir linki (Hat A), içinde bir CIC bulunan Başlangıç Adres Mesajı (IAM, Initial Adress Message) gönderek yakalar ve bu IAM transit santrale bir hat'ın yakalandığını belirtir ve istenen servis türünün özelliğini belirtir. Bu defa Geçiş Santrali (TE, Transit Exchange) başka bir IAM'ı başka bir hat'ı tutmak için hedef yerel santrale gönderir ve başka bir CIC tahsis eder. Son durumda hedef yerel santral D kanalı üzerinden hedef TE'ye bir kurulum (SETUP) mesajı gönderir.

Hedef lokal santral uzak terminalden Q.931 çalıyor (ALERTING) mesajını aldığı zaman aramayı başlatmak için gerekli tüm bilginin tamam olduğunu bilir. Yerel santral bir adres tamam mesajı (ACM, Adress Complete Message)'ını geçiş santraline gönderir ve bu defa geçiş santrali de başlangıç yerel santraline bir ACM gönderir. Aranan taraf telefona cevap verdikten sonra uzak TE D kanalı üzerinden bir Q.931 bağlantı (CONNECT) mesajını gönderir ve hedef yerel santralin cevap mesajı (ANM, Answer Message) göndermesine neden olur.

Bu noktada uç kullanıcılar arasındaki haberleşme başlangıç ve hedef ISDN erişim arayüzlerindeki ISDN B kanalları ve ağdaki A ve B hatları üzerinde oluşur. Arama çözümü Q.931 bağlantı koptu (DISCONNECT) mesajı ile başlatılır ve SS7 ağında gönderilen bırakma (REL, Release) ve bırakma tamamlandı (RLC, Release Complete) mesajları ile tamamlanır.

ISUP işaretleşme işlevleri ISDN ilave servislerin bazılarını da destekleyebilir:

- Çağrı aktarma (Call forwarding)
- Arayan Abonenin Numarasının Gösterilmesi (Calling line idendification)
- Kapalı Kullanıcı Grubu (Closed user group)
- Direkt Dahili Numara Çevirebilme (Direct dialing in)
- Kullanıcıdan Kullanıcıya İşaretleşme (User-to-user signalling)

2.3.5. İşlem yetenekleri uygulama bölümü

TCAP SS7 için genel maksatlı uzak işlem fonksiyonları sağlar. Uzak bir düğüm noktasında bir işlem gerçekleştirerek sonuçların başka bir düğüm noktasından alınmasını sağlar (Şekil 2.5).

TCAP orjinalde veri tabanları üzerinde sorgulama yapma ve sonuç alma maksadıyla tasarlanmıştır. OSI'nın uygulama katmanına denk gelir. Örnek

olarak çok büyük bir turizm acentası düşünelim. Bir çok yerde çok değişik hizmetler veren bu acentanın rezervasyon birimleri değişik yerlerde farklı reservasyon hizmetleri versin. Ankara'dan havayolu için, İzmir'den Demiryolu için reservasyon isteği yapılabilsin. Bir kullanıcı rezervasyon için diyelimki 800'ü çevirdi. Bu 800'le başlayan numaranın devamında hangi ulaşım vasıtasından reservasyon yapılması istendiği santralde kayıtlı bir anons vasıtasıyla sorulur. Kullanıcı ilave numara çevirerek tam isteğini belirtir. Bu şekilde yönlendirme tamamlanır. Bu esnada başka mesajlar ve bileşenler kullanılabilir .

2.3.6. İşletim bakım ve yönetim bölümü

OMAP (Operations Maintenance and Administration Part) SS7 ağının merkezi olarak yönetimi ve gözetimi için gerekli uygulama protokollerini ve presüdürleri sağlar. OMAP OSI'nın Ağ yönetimi protokolü olan CMIP'a yakındır. OMAP prosüdürleri bir çok fonksiyon için kullanılır:

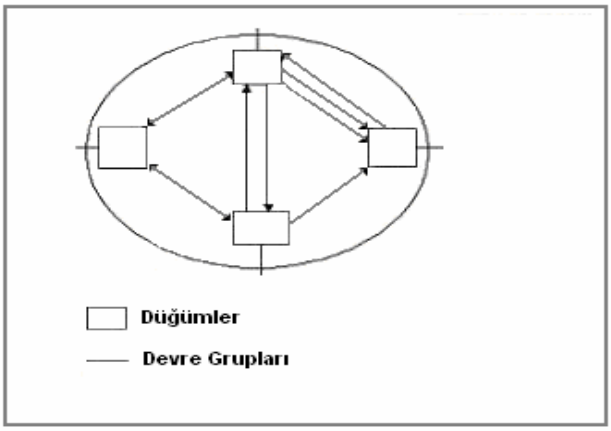
- MTP yön doğrulaması, yönlerdeki kısır döngülerin, aşırı gecikmelerin veya sinyal noktalarının ulaşılabilirliğinin tespiti için,
- SCCP yön doğrulaması, yönlendirmenin ve genel başlığın çevrilmesi için,
- Yönlendirme verisinin yönetimi,
- Trafik verisinin toplanması,
- Link arıza yönetimi.

2.4. Yönlendirme

Trafik yönlendirmesinin amacı, ağ içerisindeki herhangi iki santral arasında en fazla sayıda başarılı bağlantının kurulmasıdır. Trafik yönlendirmesinin fonksiyonu, ağ içindeki bir santralde verilen bir talebi veya trafik akışı için özel bir devre grubunun seçimidir .

Bir başlangıç noktasından hedef noktasına yapılan arama, sırasıyla gidiş ve dönüş yönünde bir çift, simetrik, noktadan noktaya tek yönlü iletişim kanalından oluşur. Hat grupların iç içe girmiş yapısı başlangıç ve hedef noktaları arasında, yönlendirme olarak isimlendirilen bağlantıyı sağlar. Her bir Başlangıç Noktasından Hedef Noktasına olan çiftin (OD, Originating Node To Destination) başlangıç santrali ve hedef santrali için bir veya daha fazla trafik akışı tanımlaması olabilir .

Bir ağ, devre grupları ile birbirine bağlanmış belirli sayıdaki santrallerden (anahtarlama merkezlerinden) oluşur. Bir santral çifti arasında birkaç direkt devre gurubu olabilir ve bunlar tek yönlü veya iki yönlü olabilirler. Şekil 2.7. olası birkaç durumu göstermektedir .



Şekil 2.7. Yönlendirme

Bir direkt yön, komşu santralleri bağlayan, bir veya daha fazla devre grubundan oluşur. Direkt olmayan yön ise diğer santraller üzerinden uçtan uca bağlantı sağlayan iki santral arasındaki devre grupları serisidir .

Genellikle bir trafik akışı için birden fazla yönlendirme kullanılır. Yön grubu ve bağlantının kurulması için gerekli yönün bu yön grubu içinden seçilme kuralları trafik akışının yönlendirme yapısını oluşturur. Yönlendirmenin bir

sonucu olarak arama, ya bir yön üzerinden kurulur ya da ağ içerisindeki kaynakların yetersizliği nedeniyle engellenir [7].

Hata veya aşırı yük durumları, yönlendirme yapıları veya yönlendirme algoritmalarının geçici olarak değiştirilmesini gerektirebilir. Bu durum bir ağ yönetim faaliyeti olarak düşünülecektir ve Uluslararası Telekomünikasyon Birliği, (ITU, International Telecommunication Union) E.400 Serisi Tavsiyeleri içinde tanımlanmıştır .

3. AĞ TOPOLOJİSİ

Bir ağ planlayıcısı açısından bakıldığında SS7 ağı aşağıdaki bileşenlerden oluşur :

- Santrallerde (düğümlerde) bağlantılı olarak yerleştirilmiş sinyalleşme noktaları
- Santrallerden izole edilmiş yüksek kapasitelerde yükü kaldırabilen sinyalleşme geçiş noktaları
- Sinyalleşme noktalarını birbirine bağlayan sinyalleşme linkleri

3.1. Ağ Bileşenleri

SS7 ağlarında, ağ topolojisi için üç tip düğüm noktası kullanılmaktadır. Düğüm noktalarının birbirlerine bağlantısı için bir sinyalleşme linki kullanılmaktadır [8]:

- Sinyalleşme uç noktası (SEP, Signalink End Point),
- Sinyalleşme Geçiş noktası (STP, Signalink Transfer Point),
- Birleşik Sinyalleşme Noktası (SEP ve STP nin birleşimi STEP),
- Sinyalleşme Linki (SL, Signalink Link).

3.1.1. İşaretleşme uç noktası SEP

Sinyalleşme uç noktası, bir haberleşme ağında, işaretleşme mesajlarını (MSU, Message Signal Unit) SS7 üzerinden bir santralden diğerine geçiren düğüm merkezleridir. Başka bir ifadeyle ağ üzerinden taşınan işaretleşme arasında geçişi SS7 ‘nin SCCP katmanında bir MSU nun DPC kullanılarak yönlendirme etiketinin oluşturulması ile gerçekleştirilir [8].

3.1.2. İşaretleşme geçiş noktası STP

STP' nin işlevi yalnızca MTP' nin özelliklerini kullanarak sinyalleşme mesajı içinde taşınan DPC ye göre bir link üzerindeki sinyalleşme mesajını alıp bunu başka bir link üzerine aktarmaktadır. STP' nin işlevi kesinlikle bir sinyal başlangıç noktası veya sinyal hedef noktası oluşturmak değildir [8].

3.1.3. Birleşik işaretleşme noktası STEP

Birleşik işaretleşme noktası, SEP ve STP' nin işlevlerini birleştirir. Böylece bir SS7 ağındaki en çok kullanılan bileşeni oluşturur .

3.1.4. Sinyalleşme hattı

SS7, iki sinyalleşme noktası arasında sinyalleşme bilgisini iletmek için sinyalleşme linklerini kullanır. Sinyalleşme linkleri ile birbirine bağlanmış olan sinyalleşme noktaları komşu noktalar olarak isimlendirilir. Sinyalleşme linkleri hem varolan PCM 30 sistemleri içinde tahsis edilmiş kanallar olabilir hem de özel olarak bu amaçla kurulmuş kanallar olabilir. Bu çalışmada kullanılan bir link grubu maximum 16 tane linkten oluşmaktadır .

3.2. Sinyalleşme bağlantısı

Sinyalleşme bağlantıları, bilgiyi kullanıcı bölümü üzerinden (TUP, ISUP, INAP) birbirlerine iletebilen bütün sinyalleşme noktaları arasında vardır. Başka bir ifadeyle bilgi bir SS7 ağı içindeki bütün sinyalleşme noktaları arasında taşınmalıdır. Bu nedenle tüm noktalar arasında bir sinyalleşme bağlantısı mevcuttur.

Sinyalleşme bağlantıları hem bağlantı temelli olarak verilen hizmetler için hemde bağlantı gerektirmeyen hizmetler için kurulabilir.

3.3. Bağlantı temelli hizmetler

Bağlantı temelli hizmetler (telefon, fax vs.) link'den link'e kurulur. Bir devre bağlantısı bir santralden diğerine bölüm bölüm kurulur. Hedefe doğru giden yön üzerindeki her bir santral bağımsız olarak kendinden sonraki bölümü belirler. Yön üzerinde varılan her bir santral OPC'yi oluşturur. Varılan santralin hedefe doğru giden yöndeki belirlediği bir sonraki santral DPC'yi oluşturur .

3.4. Bağlantısız hizmetler

Bağlantı temelli hizmetlerin tersine, bir bağlantısız hizmetde devre ağının kaynakları kullanılmaz, bağlantı uçtan uca gerçekleştirilir. Bu hizmetlere örnek olarak bir sinyalleşme kontrol noktası (SCP) ücretsiz telefon numarası sorgulaması verilebilir. Bu tür bir bağlantı paket-temelli bağlantı olarak adlandırılır .

4. AĞ YAPISI

4.1. Temel Yapısal Özellikler

Seçilecek SS7 ağ yapısı aşağıdaki durumlar göz önünde bulundurularak belirlenir:

- Maliyet etkinliğinin göz önünde bulundurulması (Gerekli kalite standartlarını sağlarken ihtiyaç duyulan trafiği kaldırabilecek maximum sayıda link veya linklerin kullanılması),
- Teknik kısıtlamalar,
- Yönetimsel gereklilikler,
- Mevcut çalışan hat ağının yapısı.

Durum ne olursa olsun düşük hata olasılığı ve çok az bir gecikme zamanıyla hızlı bir şekilde mesaj alışverişi sağlanmalıdır [9].

4.2. Yapıyı Belirleyen Özellikler

SS7 ağının boyutu maliyet etkinliğinin göz önünde bulundurulması nedeniyle ve teknik sınırlamalardan dolayı yıldız tipi hiyerarşik bir yapıya ihtiyaç oluşturabilir. Üst seviyeler STEP'lerden veya büyük yükleri kaldırabilen STP'ler den oluşurken alt seviyeler tamamiyle SEP lerden oluşur. Hiyerarşik seviyelerin sayısı aşağıdaki faktörlere bağlıdır;

- Ağdaki toplam SP sayısı,
- Ağ bileşenleri için maksimum kanal ağlantı kapasitesi,
- Ağ bileşenleri için maksimum işlem kapasitesi,
- Bir sinyalleşme mesajı için maksimum kabul edilebilir gecikme zamanı,
- Varılan bir ağın veri iletim hatlarına mümkün olan en yüksek seviyede uyum sağlayabilme.

Eğer SS7 ağı yukarıda bahsedilen nedenler göz önünde bulundurularak hiyerarşik bir yıldız yapıda oluşturulacaksa planlamacının aşağıdaki şartları yerine getirmesi gerekir;

- Gerekli hiyerarşik seviye sayısının belirlemesi,
- Herbir SP'nin bir hiyerarşi seviyesine atanması,
- Farkli seviyelerdeki SP'ler arasındaki en az 2 SP'yi desteklemelidir,
- Trafiğin üst seviyelerdeki STP'ler arasında mümkün olduğunca eşit dağıtılması,
- Bir üst hiyerarşik seviyedeki 2 SP'nin servis dışı kalması durumunda sinyalleşme mesajlarının mümkün olan en az kayıba uğraması,
- Link bağlantı kapasitesi ve maksimum işlem performansı açısından SP'nin sınırlarının belirlenmesi.

Bu koşulları sağlamanın iki farklı yolu vardır. İlk strateji ağı üst seviyedeki SP çiftlerinin kalıcı olarak atanacağı hem yönetim açısından hemde coğrafi olarak bölümlere ayırır [9].

İkinci strateji SP lerin STEP lere daha örtüşümlü bir yapıda atanmasını sağlar. Öte yandan optimum ilişkilendirme devre ağındaki mevcut yollar nedeniyle büyük ölçüde zorlaşır. Aşırı yüklü STP'lerin bulunduğu bir SS7 ağında ilişkilendirmeler bütünüyle serbest olarak seçilebilir ve yalnızca maliyet etkinliği nedeniyle sınırlanır .

4.3. Sinyalleşme Noktalarının Numaralandırılması

Bir sinyalleşme ağındaki sinyalleşme noktaları mesajları (MSU) birbileri arasında iletirler. Mesajların gerekli hedeflere ulaşmasını sağlamak için hedeflere bir adres ataması yapılır. Bu adres sinyalleşme nokta kodu (SPC, Signalink Point Code) ağdaki herbir sinyalleşme noktası için ayırt edici bir özellik sağlar.

Her MSU üzerindeki sinyalleşme bilgisi alanında (SIF, Signalink Information Field) OPC ve DPC bilgilerini taşır. Adres bilgisinin uzunluğu 14 bit veya 24 bit olabilir. Bu yüzden farklı kodlar kullanılabilir.

Bir ağ içerisindeki sinyalleşme noktalarının numaralandırılmasında;

- Serbest olarak seçilmiş numaralandırma
- Bölgesel yapılara göre numaralandırma
- Sinyalleşme ağının hiyerarşik yapısına göre numaralndırma
- Devre ağındaki santrallere ait kodlar temel alınarak yapılan bir numaralandırma özellikleri kullanılabilir.

Serbest olarak seçilen numaralandırmanın dezavantajı büyük ağlarda karışıklığa neden olabilmesidir. Bölgelere göre veya hiyerarşik yapıya göre numaralandırma yapılması daha uygundur.

4.4. Yönler, Yön Setleri, Yük Dağılımı ve Varış Noktaları

Güvenlik ve yük dağılımı göz önünde bulundurulduğunda bir işaretleşme ağı içinde işaretleşme mesajlarının iletimi için birden fazla yön tanımlaması yapılmalıdır. Yönler sırasıyla ilerleyen bir veya birden fazla linkten oluşur. Yönlendirme, işaretleşme mesajlarının seçilen yönlere göre hedefe doğru gidiş yönünün tarif edilmesidir .

Bir SS7 ağında yönlendirme SS7 katmanlarından mesaj iletim bölümünün (MTP katman 3) bir servisidir. Ağ içindeki her bir sinyalleşme noktası yönlendirmenin kendinden sonraki kısmını birbirlerinden bağımsız olarak bağlantısı olan kanalların çalışmasına ve kendi üzerindeki yönlendirme tablosuna göre yerine getirilir .

4.5. SS7 Ağları için Genel Yönlendirme Kuralları

Bir yön grubu içindeki yönler tanımlanırken aşağıdaki kurallara uyulmalıdır ;

- Yönün uzunluğu başlangıç SP'si ile hedef SP arasındaki STP sayısı, sinyalleşme mesajlarının iletim zamanını en aza indirmek için mümkün olduğu kadar kısa tutulmalıdır,
- Eğer iki nokta arasında direk bağlantı bulunuyorsa, bu link daima yönlendirme grubu içinde standart yön olmalıdır.
- Bununla birlikte yük dağılımı yapılarak 2 tane standart yön tanımlanabilir. Her iki yönünde uzunluğu aynı olmalıdır.
- Farklı uzunluktaki yönler için, yük dağılımı yapılmasından kaçınılmalıdır.
- Yönler sinyalleşme mesajlarının döngüye girmesini engelleyecek şekilde oluşturulmalıdır [9].

5. UYGULAMA

5.1. Programın Genel Tanıtımı

Gelişen anahtarlama teknikleri, iletim ortamı teknolojisi ve sayısal teknoloji ile birlikte, Servis Sağlayıcılar çok daha fazla sayıda abonenin daha uzak mesafelerde kesintisiz olarak haberleşmesini sağlamayı amaçlamaktadırlar. Teknolojinin bugünkü geldiği noktada haberleşme çok küçük hata oranları ile sağlanabilmektedir. Ancak doğal afet, alt yapı çalışmaları ve operatör hataları gibi önceden öngörülemeyen arızalar, özellikle statik yönlendirme kullanılan ağlarda geçici süreli kesintilere neden olmaktadır.

Statik yönlendirmenin kullanılacağı bir ağ tasarımında yönlendirmenin planlama aşaması büyük önem taşımaktadır. İyi planlanmamış bir yönlendirme uygulamasında yukarıda bahsedilen nedenlerle oluşacak bir hata, ağ üzerinde kısmi tıkanmalara ve döngülere neden olur. Aynı zamanda günümüzdeki rekabet ortamı içinde sunulan servislerden kazanç sağlanabilmesi için servis sağlayıcılar, kuracakları ağ için maliyet hesaplamaları yapmak zorundadırlar. Hizmet verilen her kullanıcı için ağ içinde belirli bir kaynak tahsisi yapılması gerekir. Yetersiz kaynak tahsisi kullanım aşamasında müşteriye sunulan hizmetlerde kesintiye neden olabilir. Düzgün yapılmamış bir planlama nedeniyle gereğinden fazla kaynak tahsisi servis sağlayıcılar için para kaybına neden olacaktır. Bu çalışmanın amacı da, kurulması planlanan ağ için döngüsel çağrı kontrollü ve kaynakları en uygun şekilde kullanacak yönlendirme tablolarının oluşturulmasıdır. Bu amaca yönelik olarak ISDN ağlar için bir Ağ Planlama programı hazırlanmıştır.

Kodların yazılması için Delphi Builder kullanılmıştır. Kodlar EK-1'de verilmiştir. Oluşturulan yönlendirme tabloları kullanıcı arayüzü üzerindeki butonlar ve kısa yol menüleri ile kullanıcıya gösterilmektedir. Aynı zaman da

yönlendirmelerin nasıl bir yol izlediği hazırlanan simülasyon uygulması sayesinde görülebilmektedir.

5.2. Bağlantı Linkleri Üzerindeki Trafik Yükünün Hesaplanması

Kurulan bir haberleşme ağında oluşturulan düğüm santralleri belirli bir abone kapasitesine sahiptir ve bu abonelerin ağ üzerinde oluşturacağı bir trafik yükü vardır. Bir düğüm santralinin ağ üzerinde oluşturacağı trafiğin hesaplanması için bazı parametreler gereklidir. Bu parametreler ağın planlama aşamasında tahmini olarak veya daha önceden elde edilen ölçüm sonuçları kullanılarak elde edilebilir. Mevcut bir ağ da ise belirli periyotlarla alınan ölçümler sonucu bu parametreler elde edilebilir [10];

CN (Call Number) : Belirli bir periyot içindeki toplam çağrı sayısını,
CD (Call Duration) : Belirli bir periyot içindeki toplam çağrı süresini (sn),
MP (Measurament Periyod) : Temel alınan zaman dilimini (sn),
ICR (Incoming Call Rate) : Düğüme gelen çağrı oranını ifade etmektedir.

Bu parametreler kullanılarak bir düğümde oluşan trafik yükü $Y_{düğüm}$ (earlang);

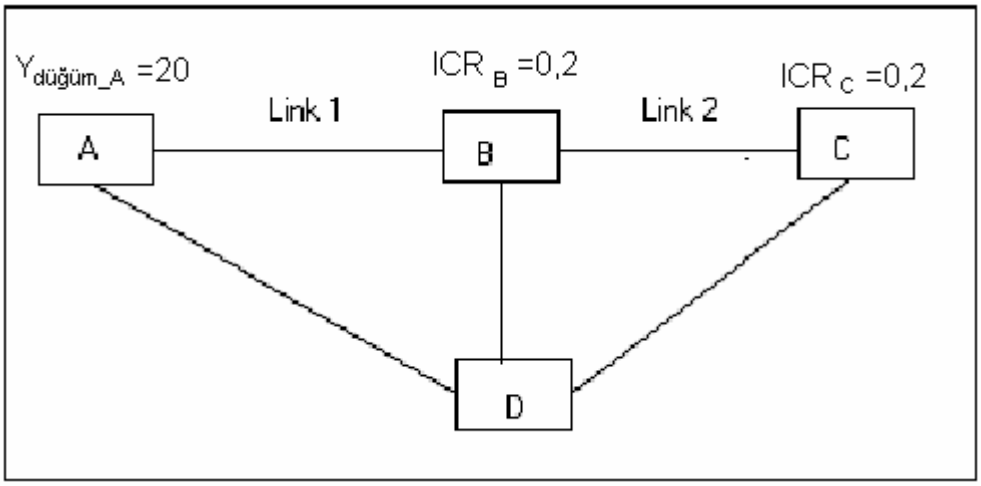
$$Y_{düğüm} = \frac{MP \,.\, CN}{CD} \qquad (5.1)$$

olarak hesaplanır.

Bu durumda düğümdeki yükün başka bir düğüme doğru giderken linkler üzerinde oluşturacağı yük Y_{Link} (earlang);

$$Y_{Link} = Y_{düğüm} \,.\, ICR \qquad (5.2)$$

Y_{Link}, oluşturulan yönlendirme tablosuna göre başlangıç düğümünden hedef düğümüne doğru geçilen her bağlantı linki üzerinde bir yük oluşturur.



Şekil 5.1. Düğümler arası trafik yükünün dağılımı

Şekil 5.1' de görüldüğü gibi bir ağ yapısını ele alalım. A düğümünden B düğümüne gidiş yön tablolarına göre Link1 üzerinden olacaktır. A düğümünden C düğümüne gidiş ise sırasıyla Link1 ve Link2 üzerinden olacaktır. Bu durumunda A düğümünde oluşan $Y_{düğüm}$ yükünün linkler üzerine paylaşımı şu şekilde olur;

A dan B' ye gidiş nedeniyle Link1 üzerindeki yük Y_{Link1}, Eş. 5.2'den;

$$Y_{Link1_AB} = Y_{düğüm_A} \cdot ICR_B = 20 \cdot 0{,}2 = 4$$

A dan C ye gidiş nedeniyle Link1 ve Link2 üzerindeki yükler aynı şekilde;

$$Y_{Link1_AC} = Y_{düğüm_A} \cdot ICR_C = 20 \cdot 0{,}2 = 4$$

$$Y_{Link2_AC} = Y_{düğüm_A} \cdot ICR_C = 20 \cdot 0{,}2 = 4$$

Bu durumda Link1 üzerindeki yük, A dan B ye gidiş nedeniyle oluşan yükle A dan C ye gidiş nedeniyle oluşan yükün toplamı olacaktır ;

$Y_{Link1_Top} = Y_{Link1_AB} + Y_{Link1_AC}$

$Y_{Link1_Top} = 4 + 4$

$Y_{Link1_Top} = 8$

olarak bulunur.

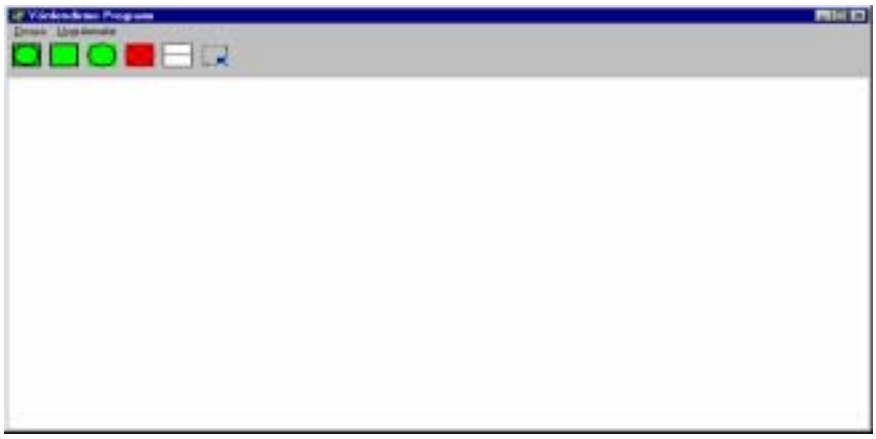
Düğümler arası bağlantılar iki yönlü olarak çalıştığı için C den A ya ve C den B ye gidişler de C düğümü üzerinde oluşan toplam yüke göre Link1 ve Link2 üzerinde yük oluşturacaktır.

Yön tabloları ağ üzerinde oluşturulan tüm düğümlerden diğer bütün düğümlere gidiş şeklinde bir matris oluşturduğu için Linkler üzerindeki yük oluşturulan yön tablolarına göre bütün düğümler için hesaplanmalıdır. Bu çalışmada geliştirilen uygulama programı yüklerin hesaplamasını yön tablolarının oluşturulmasının ardından otomatik olarak yapmaktadır. Linkler üzerindeki yük hesaplanırken başlangıç veya bitiş noktası olarak STP tipi düğümler göz ardı edilmiştir. STP tipi düğümlerden çağrı başlangıcı veya çağrı bitişi yapılamaz çünkü bu tip düğüm noktaları yalnızca geçiş noktası olarak kullanılabilir .

5.3. Kullanıcı Arayüzü

Kullanıcı ara yüzü (Şekil 5.2), düğüm noktalarının istenilen herhangi bir yere konulabileceği beyaz zeminli boş bir uygulama alanı, ağ içinde kullanılacak düğüm tiplerinin ve santral bağlantılarının seçilebileceği bir buton alanı, uygulamaların yapılabileceği bir menü alanından oluşmaktadır. Butonlar farklı renk ve şekiller kullanılarak oluşturulmuştur. Her bir buton farklı bir tip düğüm noktasının oluşturulması için kullanılmaktadır.

Menü alanı programın kullanımı başlığı altında anlatılacaktır.

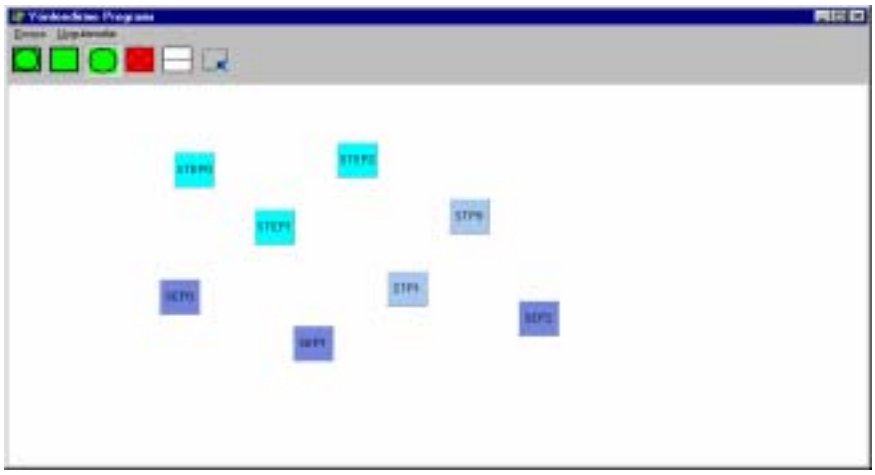


Şekil 5.2. Kullanıcı arayüzü

Menü alanı içinde Yönlendirme tablolarının oluşturulması, simülasyon ve yük dağılımlarının hesaplanması uygulamaları bulunmaktadır.

5.3.3. SEP butonu

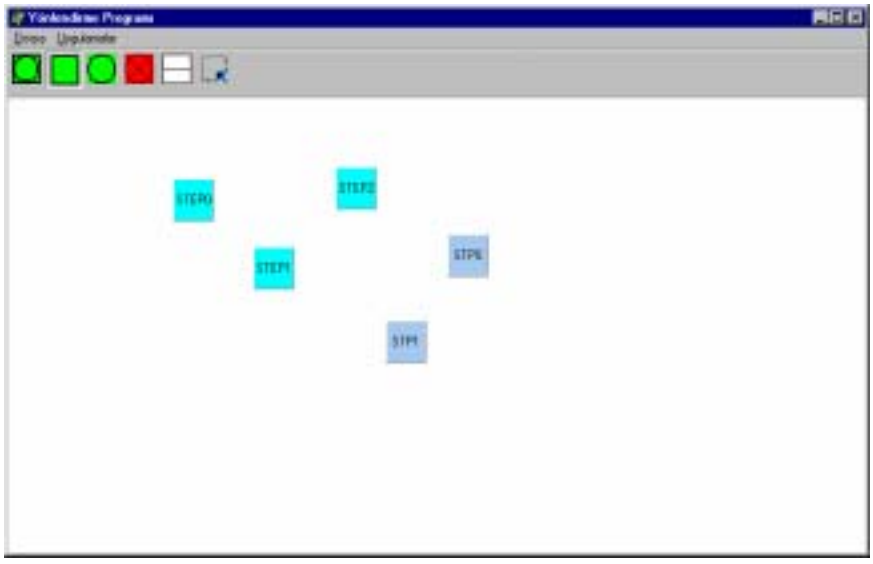
Kullanıcı bu butonu mouse ile tıkladığında buton seçili halde kalacaktır. Buton seçili iken uygulama alanı üzerindeki herhangi bir yer mause ile tıklandığında o noktaya bir adet SEP düğüm noktası oluşturulacaktır. Butonlar ‘SEP0’, ‘SEP1’ ve artan rakamlarla isilendirilmektedir. Buton oluşturulduğunda koyu mavi renkle gösterilir. Şekil 5.3’de SEP butonunun kullanımı görülmektedir.



Şekil 5.3. SEP butonunun kullanımı

5.3.2. STP butonu

Kullanıcı bu butonu mouse ile tıkladığında buton seçili halde kalacaktır. Buton seçili iken uygulama alanı üzerindeki herhangi bir yer mause ile tıklandığında o noktaya bir adet STP düğüm noktası oluşturulacaktır.

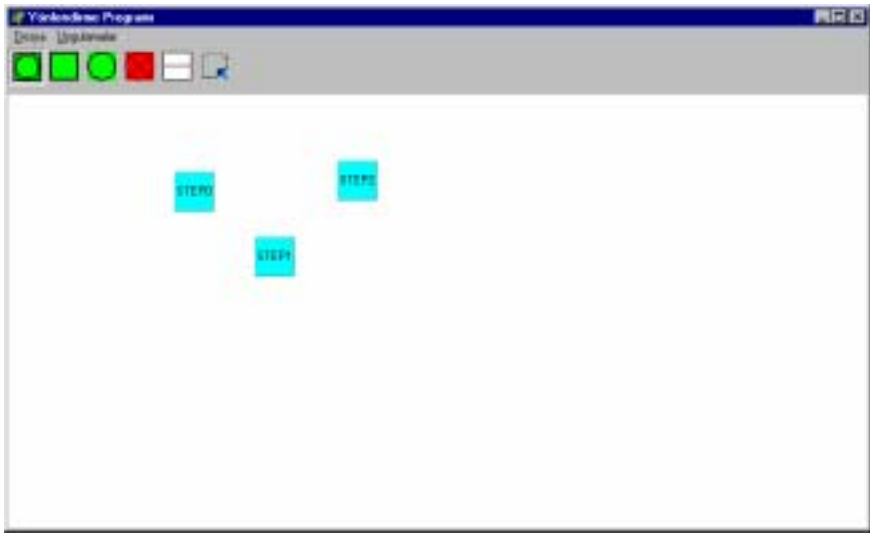


Şekil 5.4. STP butonunun kullanımı

Butonlar ‘STP0’, ‘STP1’ ve artan rakamlarla isimlendirilmektedir. Buton oluşturulduğunda mavi renkle gösterilir. Şekil 5.4’de STP butonunun kullanımı görülmektedir

5.3.1. STEP butonu

Kullanıcı bu butonu mouse ile tıkladığında buton seçili halde kalacaktır. Buton seçili iken uygulama alanı üzerindeki herhangi bir yer mause ile tıklandığında o noktaya bir adet STEP düğüm noktası oluşturulacaktır. İlk oluşturulan düğüm "‘STEP0’ bir sonraki düğüm ‘STEP1’ ve her yeni düğüm artan rakamlarla isimlendirilir. Buton oluşturulduğunda açık mavi renkle gösterilir (Şekil 5.5).

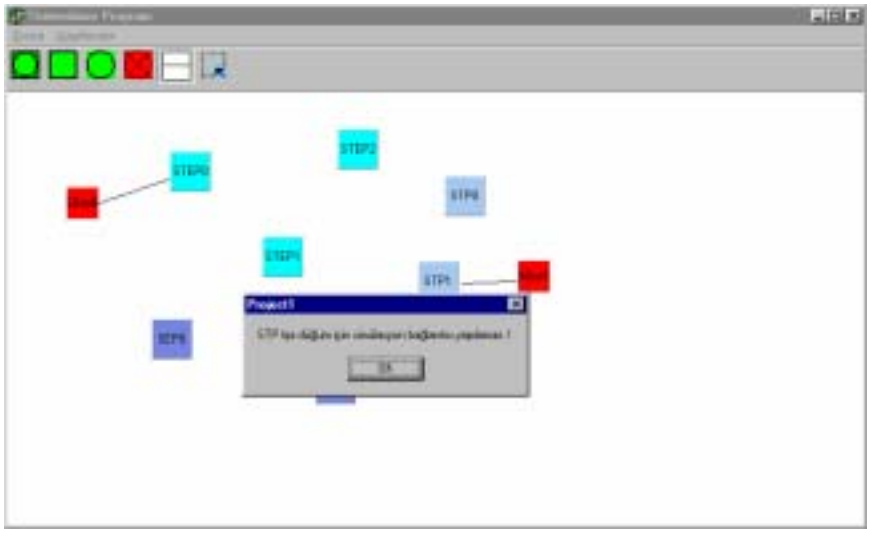


Şekil 5.5. STEP butonunun kullanımı

5.3.4. Simülasyon butonu

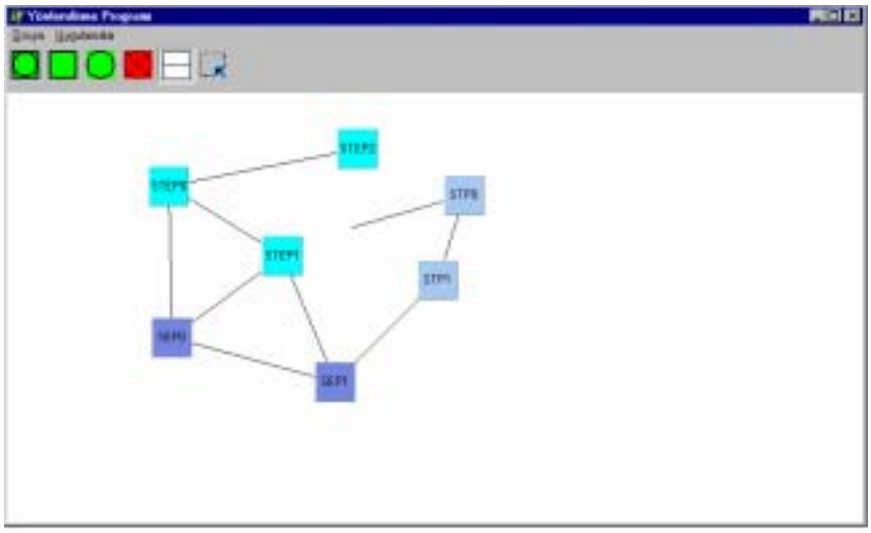
Bu buton seçildiğinde uygulama alanı üzerine Simülasyon Kutuları oluşturulmaktadır. En fazla iki adet kutu oluşturulabilir. Kutular yönlendirmenin izlenmek istendiği düğümlere bağlanması gerekmektedir. Oluşturulan ilk kutu ‘SBox0’ ikinci kutu ‘SBox1’ olarak isimlendirilir. Kutular

STEP tipi veya SEP tipi düğümlere bağlanmalıdır. STP tipinde bir düğüme bağlanmak istediğinde kullanıcı bir mesajla uyarılmaktadır (Şekil 5.6).



Şekil 5.6. Simülasyon butonunun kullanımı

5.3.5. Link butonu

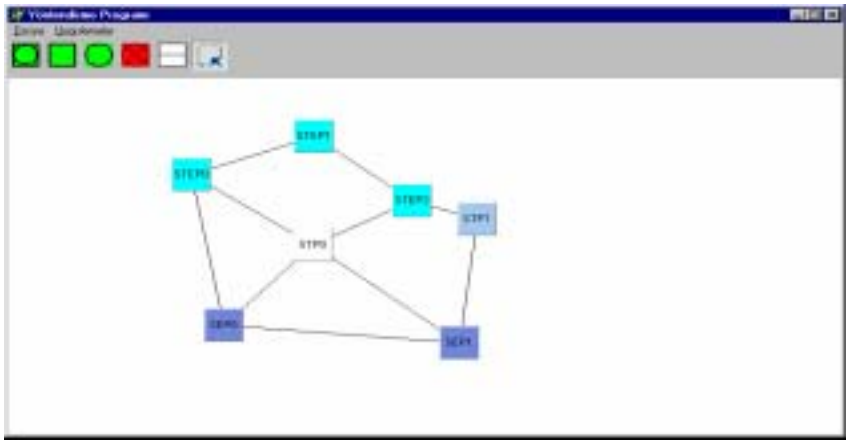


Şekil 5.7. Link butonunun kullanımı

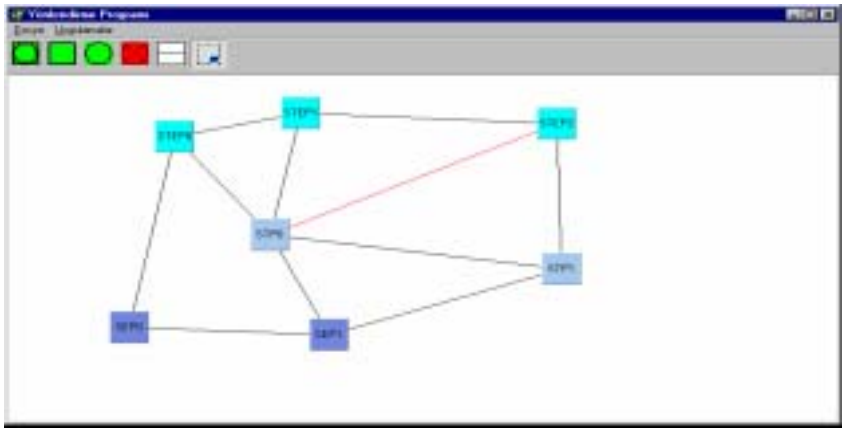
İkinci bir düğüme tıklandığında bağlantı tamamlanmış olur. Link oluşturulduğunda siyah renkle gösterilir (Şekil 5.7).

5.3.6. Seçim butonu

Buton seçildikten sonra uygulama alanı üzerinde önceden yaratılmış olan nesneler tıklanarak seçili hale getirilebilir. Seçilen nesne bir düğüm noktası ise (Şekil 5.8), tipi ne olursa olsun beyaz renkli hale gelmektedir. Bu düğümün sürüklenerek yeri değiştirilebilir veya düğüm sağ tıklanarak silinebilir.



Şekil 5.8. Seçim butonunun düğümler için kullanımı



Şekil 5.9. Seçim butonunun linkler için kullanımı

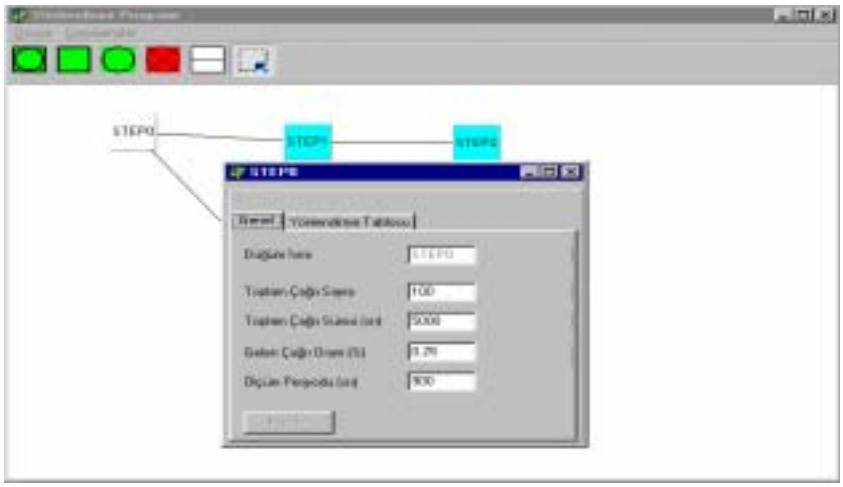
Seçilen nesne bir bağlantı linki ise (Şekil 5.9), rengi kırmızıya dönüşecektir. Seçili link sağ tıklanarak silinebilir.

5.3.7. Yardımcı menüler

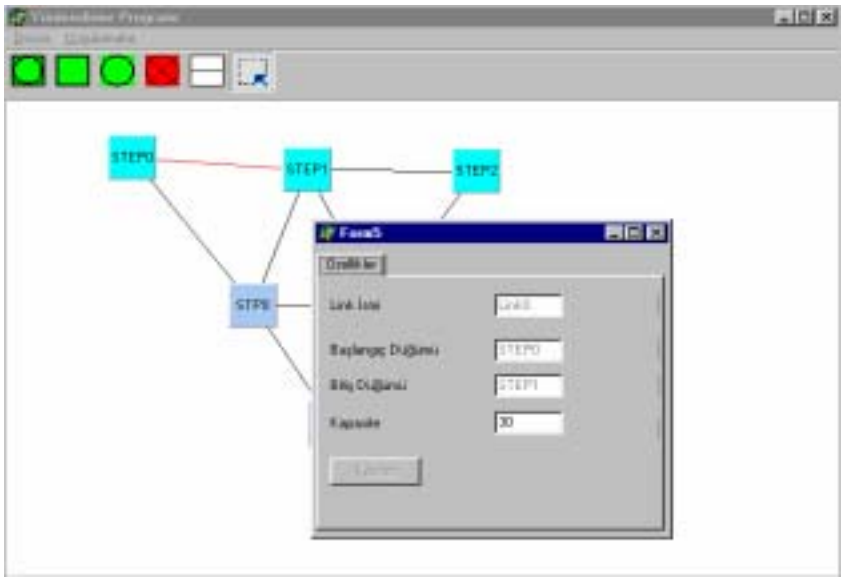
Uygulama alanı üzerinde oluşturulan düğümler veya linkler mouse'un sağ butonu ile seçildiğinde açılan yardımcı menüler üzerinde 'Özellikler' seçilerek seçili nesneye ait özellikler görüntülenebilir.

Seçili nesne bir düğüm noktası ise düğüm noktasının adı ve trafik parametreleri görülebilir. Trafik parametreleri kullanıcı tarafından değiştirilebilir (Şekil 5.10).

Seçili nesne bir bağlantı linki ise linkin adı linkin birbirine bağladığı başlangıç ve bitiş düğümlerinin adı ve link kapasitesi görülebilir. Link kapasitesi kullanıcı tarafından değiştirilebilir (Şekil 5.11).



Şekil 5.10. Düğüm noktasının özelliklerinin görüntülenmesi

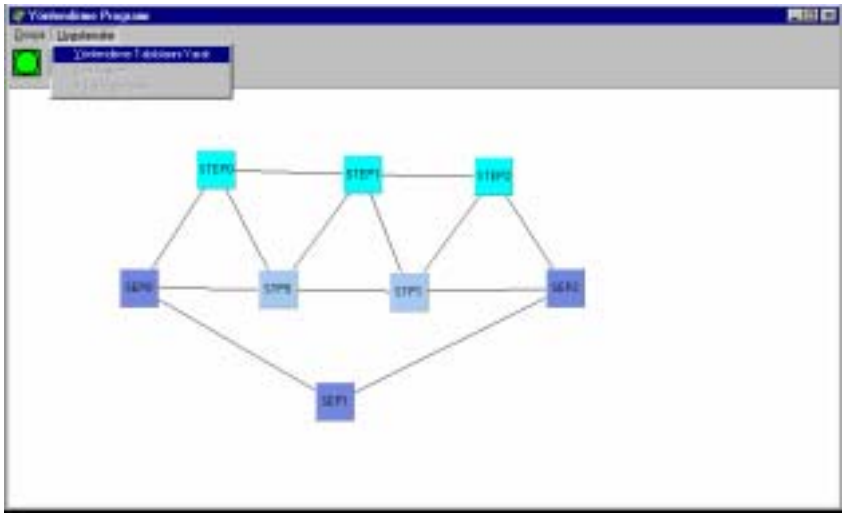


Şekil 5.11. Bağlantı linkinin özelliklerinin görüntülenmesi

5.4. Programın Kullanımı

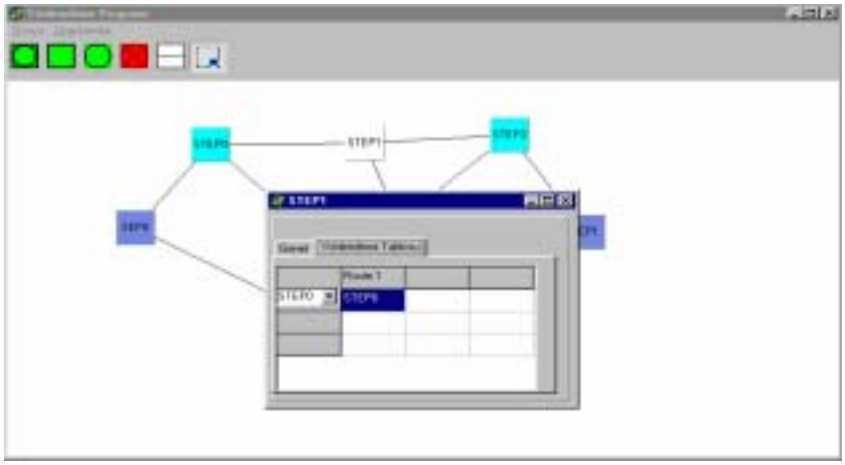
Programın çalıştırılması ile kullanıcı ekranına boş bir uygulama alanı gelmektedir. Bu alan ağın oluşturulacağı coğrafik bölgeyi temsil etmektedir. Kullanıcı istediği düğüm tipini buton alanınından mause'la tıklayarak seçtikten sonra uygulama alanı üzerinde istediği noktaya yine mause'la tıkladığında seçtiği düğüm tipi o nokta üzerinde oluşturulur. Düğümler uygulama alanı üzerine yerleştirildikten sonra buton alanı üzerinden bağlantı linki seçilerek istenilen düğümler birbirlerine bağlanır. Üzerine ilk tıklanan düğüm başlangıç düğümünü oluşturur. Mause butonu bırakılmadan hareket ettirildiği süre boyunca link uygulama alanı üzerinde çizilecektir. Bu aşamada başka bir düğüm üzerine tıklandığında bağlantı bu iki düğüm arasında oluşturulacak ve bu link veritabanına işlenecektir. Ağ oluşturulduktan sonra kullanıcı silmek istediği düğüm yada bağlantı linkini, mause'la sağ tıklayıp açılan yardımcı menüdeki sil kısmını kullanarak silebilmektedir. İstenildiği takdirde buton alanı üzerindeki 'Seçim' butonu tıklandıktan sonra herhangi bir düğüm uygulama alanı üzerinde başka bir noktaya sürüklenerek götürülebilmektedir. Programın bu özellikleri kullanılarak istenilen ağ uygulama alanı üzerinde oluşturulabilmektedir.

İstenilen ağ oluşturuldaktan sonra ana menü kısmındaki ‘uygulamalar’ alanı içinden ‘Yönlendirme Tablolarını Oluştur’ kısmı seçilerek tablolar oluşturulabilmektedir (Şekil 5.12).



Şekil 5.12. Yönlendirme tablolarının oluşturulması

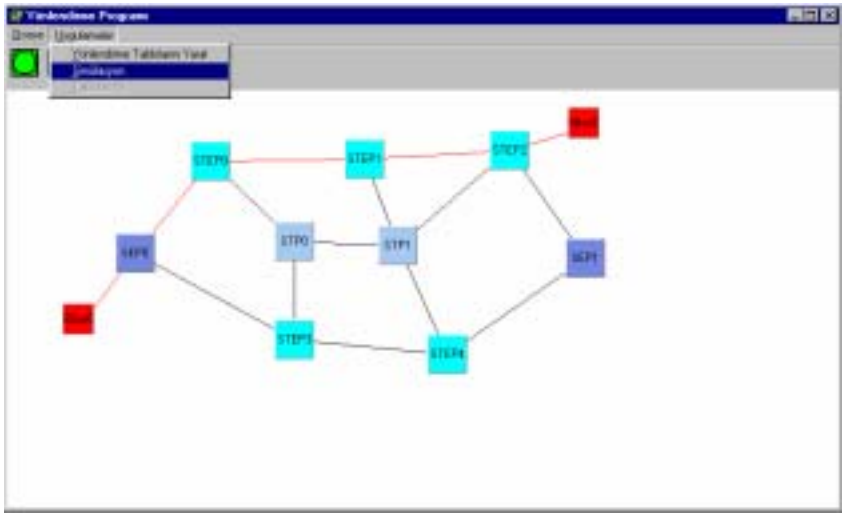
Tabloların oluşturulmasının ardından düğüm üzerinde sağ tıklanarak açılan yardımcı menü üzerinden o düğüme ait tablo görebilir (Şekil 5.13).



Şekil 5.13. Yönlendirme tablolarının görüntülenmesi

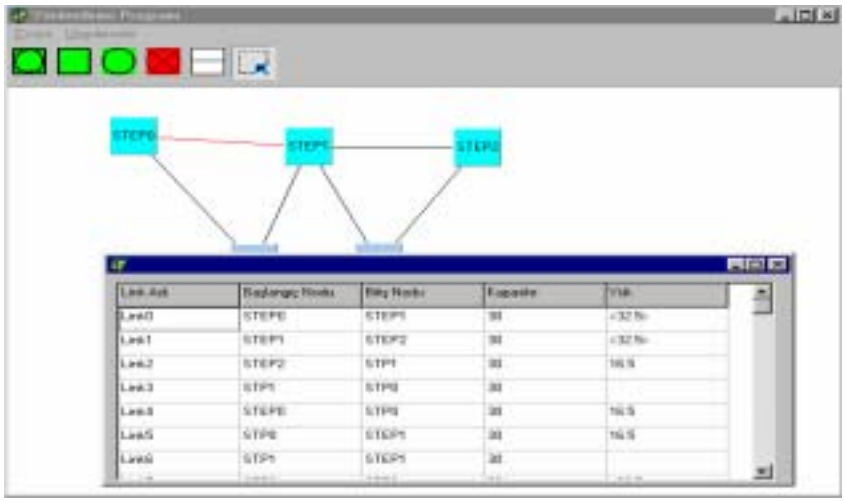
Tabloların oluşturulmasının ardından yine ana menü içindeki simülasyon kısmı aktif hale gelmektedir. Kullanıcının bu kısmı kullanabilmesi için, önceden iki adet simülasyon kutusunu iki farklı düğüm noktasına bağlaması gerekmektedir (Şekil 5.14). Simülasyon kutuları düğüm noktalarının bağlanması mantığı ile aynı olarak uygulama alanı üzerine yerleştirilmektedir. Simülasyon kutularının bağlandığı düğümler arası yönlendirme tablolarının izlediği yol kırmızı renkle görülebilir.

Kullanıcının herhangi bir düğüm yada bağlantı linkini silmesi durumunda tablolar yeniden oluşturulmalıdır.



Şekil 5.14. Yönlendirmelerin simülatör kutusu ile görüntülenmesi

Yönlendirme tablolarının oluşturulmasının ardından menü alanındaki uygulamalar kısmı içindeki ‘Yük hesaplama’ seçilerek bütün bağlantı linkleri üzerindeki yük durumu görüntülenebilir (Şekil 5.15).



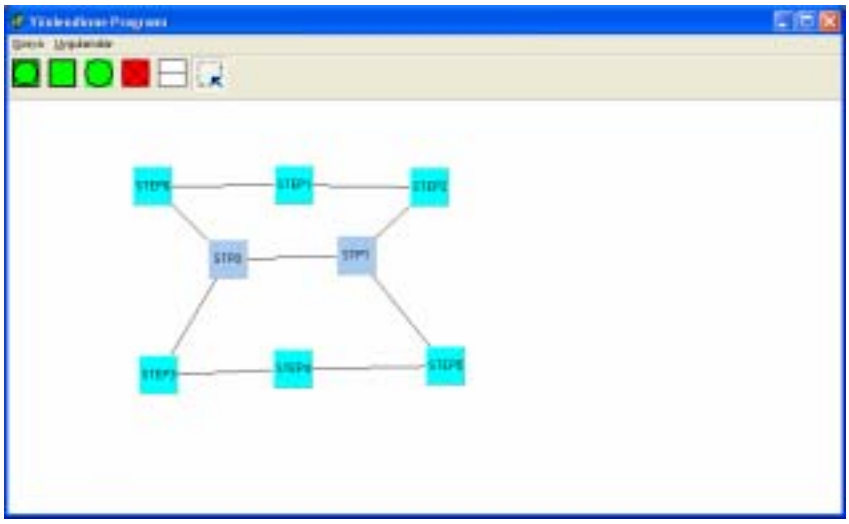
Şekil 5.15. Bağlantı linkleri üzerindeki yük durumlarının görüntülenmesi

Şekildeki yük tablosunda mevcut bütün linklere ait kapasite ve yük durumu görülmektedir. Kapasite miktarını aşan yük '**< >**' işaretleri arasında gösterilerek kullanıcı uyarılmaktadır. Yükün kapasiteyi aştığı linkler için kapasite kullanıcı tarafından artırılmalıdır. Yük alanı boş olan linkler üzerinde herhangi bir trafik oluşmamıştır. Oluşturulan yönlendirme tablolarına göre bu linkler gereksizdir ve kullanıcı tarafında silinmelidir.

5.5. Programın Örnek Uygulamaları

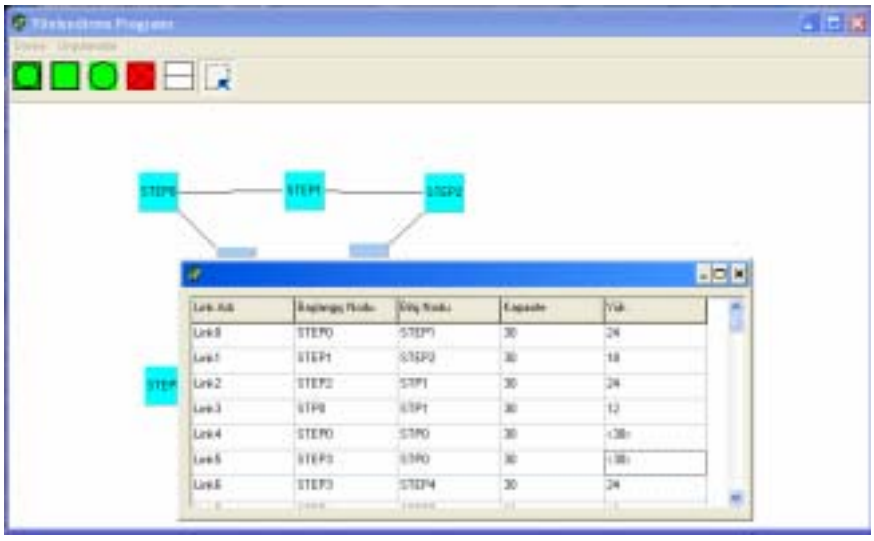
5.5.1. Uygulama 1

Şekil 5.16'da görüldüğü gibi 6 adet STEP tipi ve 2 adet STP tipi düğümden oluşan bir ağ oluşturulsun. Şekildeki ağ için yönlendirme tabloları oluşturulduktan sonra bağlantı linkleri üzerindeki yük dağılımları hesaplansın. Uygulamalar menüsünden 'Yük hesaplama' kısmı seçilerek Trafik yükünün linkler üzerinde dağılımı görülebilir.

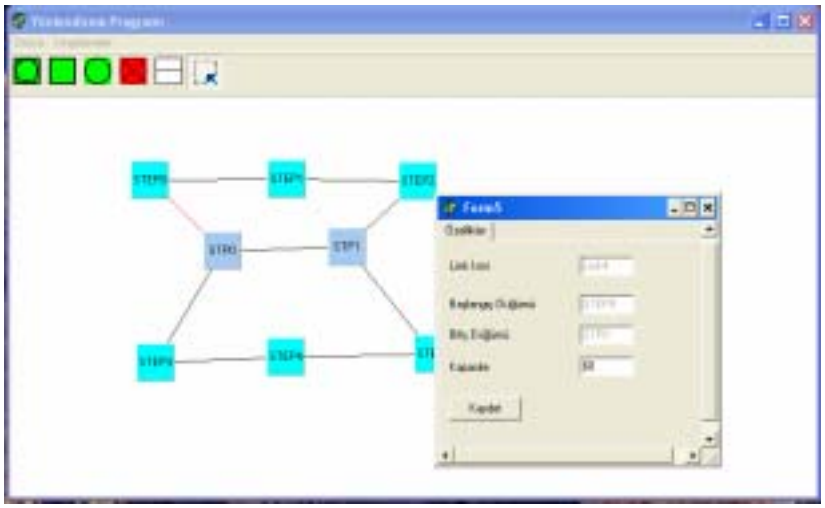


Şekil 5.16. Uygulama 1 için ağ yapısı

Şekil 5.17'de görüldüğü gibi STEP0 ve STP0 düğümleri arasında bulunan Link4 üzerindeki trafik yükü kapasite sınırındadır ve kullanıcı '< >' işaretleri ile uyarılmaktadır. Aynı şekilde STEP3 ve STP0 düğümleri arasında da trafik yükünün kapasite sınırında olduğu görülmektedir. Bu aşamada kullanıcının bu linkler üzerindeki kapasite miktarını artırması gerekmektedir.

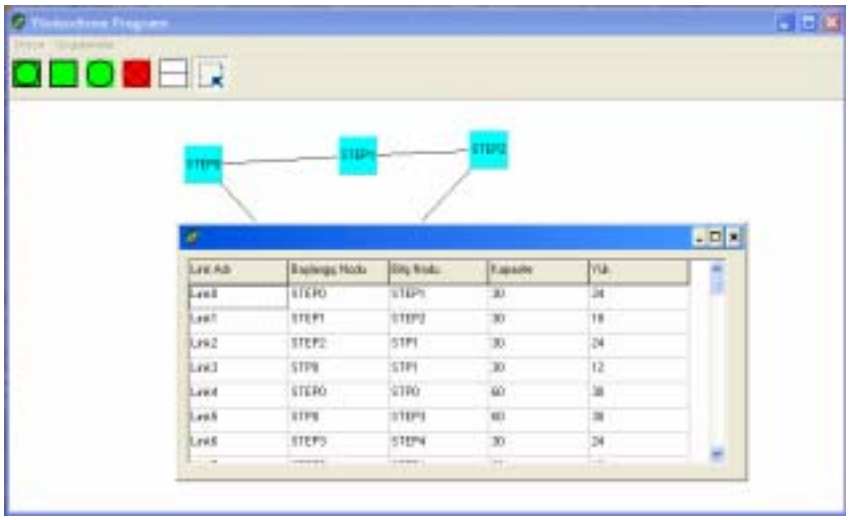


Şekil 5.17. Uygulama 1 için yük dağılımı



Şekil 5.18. Uygulama 1 için linklerin kapasitesinin değiştirilmesi

STEP0 ve STP0 düğümleri arasındaki Link4 ’ün ağ oluşturulurken 30 olarak program tarafından atanmış olan link kapasitesi 60 olarak değiştirilsin (Şekil 5.18). Aynı şekilde, daha önce yük dağılım tablosundan kapasite sınırında yüke sahip olduğu görülen STEP3 ve STP0 düğümleri arasındaki Link5 ‘in kapasitesi 60 yapılsın.



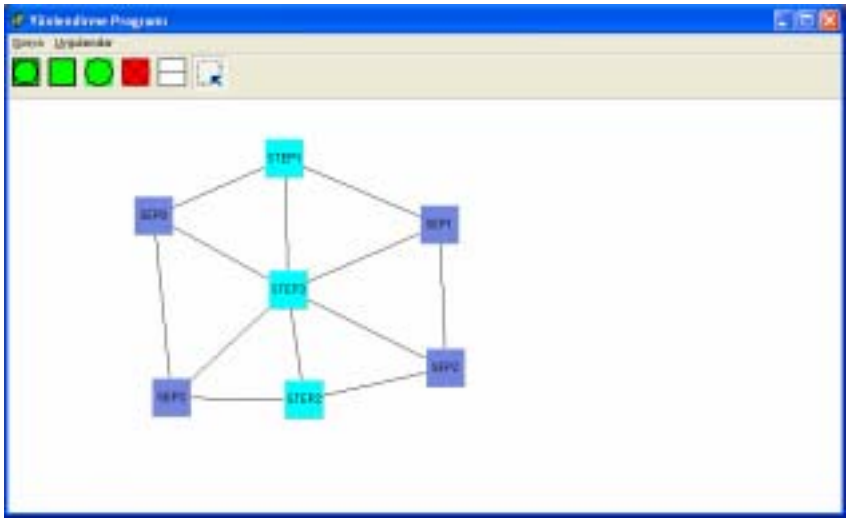
Şekil 5.19. Uygulama 1 için linklerin kapasitesinin değiştirilmesi sonrası yük dağılımı

Şekil 5.19’da görüldüğü gibi kapasitelerin artırımından sonra yük değeri üzerindeki uyarı işareti kalkmıştır. Linkler üzerinde herhangi bir değişiklik

yapılmadığı için linkler üzerindeki yük miktarı da değişmemiştir. Kullanıcı bu sayede bu şekilde bir ağ dizaynı yaparken Link4 ve Link5 bağlantılarını 60 kanal kapasiteli olarak planlayacaktır.

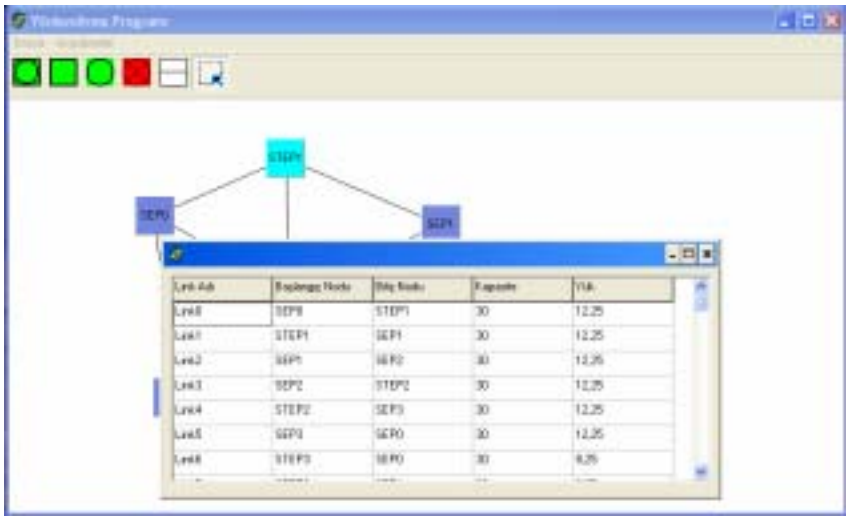
5.5.2. Uygulama 2

Bu uygulama için 3 adet STEP tipi düğüm ve 4 adet SEP tipi düğüm birbirine bağlanarak Şekil 5.20'de görülen ağ oluşturulsun. Oluşturulan bu ağ için yük dağılımı hesaplansın



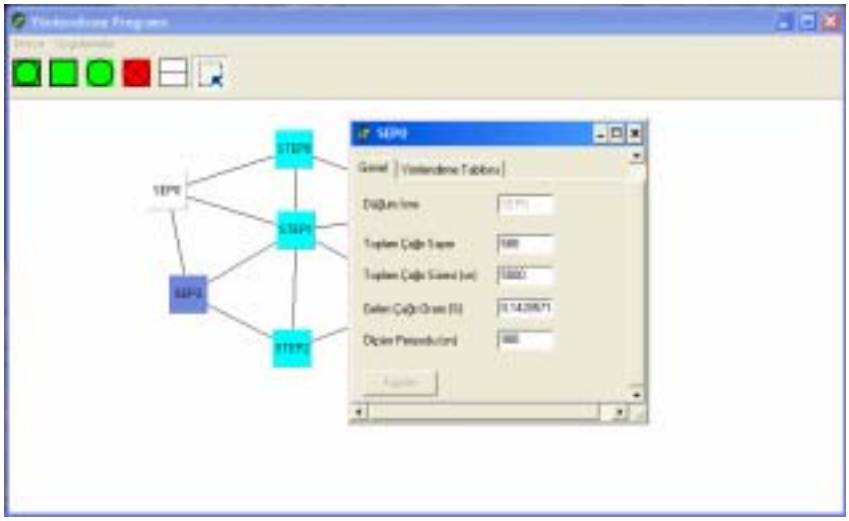
Şekil 5.20. Uygulama 2 için ağ yapısı

Yük dağılım tablosu Şekil 5.21'de görülen ağ için SEP0 düğümünün oluşturacağı yük, düğüm üzerindeki trafik parametrelerine göre belirlenir. Trafik parametreleri düğüm uygulama alanı üzerinde ilk kez oluşturulurken program tarafından otomatik olarak belirlenmektedir. Yük dağılım tablosu da bu parametrelere göre hazırlanmaktadır. Düğüm üzerindeki trafik parametrelerinin kullanıcı tarafından değiştirilmesi ile linkler üzerindeki yük dağılımları da değişecektir.



Şekil 5.21. Uygulama 2 için yük dağılım tablosu

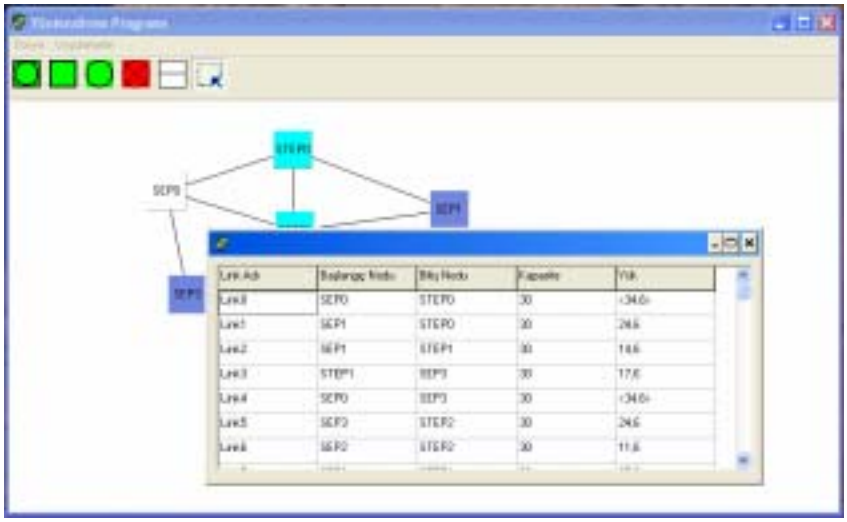
Şekil 5.22 den görüldüğü gibi kullanıcı tarafından SEP0 düğümü üzerindeki ‘Toplam Çağrı Sayısı’ parametresi başlangıçtaki 100 değerinden 500 değerine çekilsin. Bu parametrenin değiştirilmesi SEP0 düğümünün oluşturacağı toplam trafik yükünü artıracaktır.



Şekil 5.22. Uygulama 2 için düğüm üzerindeki trafik parametrelerinin değiştirilmesi

Yeni yük dağılımını görüntülemek için yük tablosu oluşturulsun. Şekil 5.23 de görüldüğü gibi linkler üzerindeki yük dağılımları değişmiş bazı linkler

üzerindeki yük kapasite miktarını aşmıştır. Bu linkler için Uygulama 1 de anlatıldığı gibi kullanıcı tarafından link kapasiteleri artırılmalıdır.

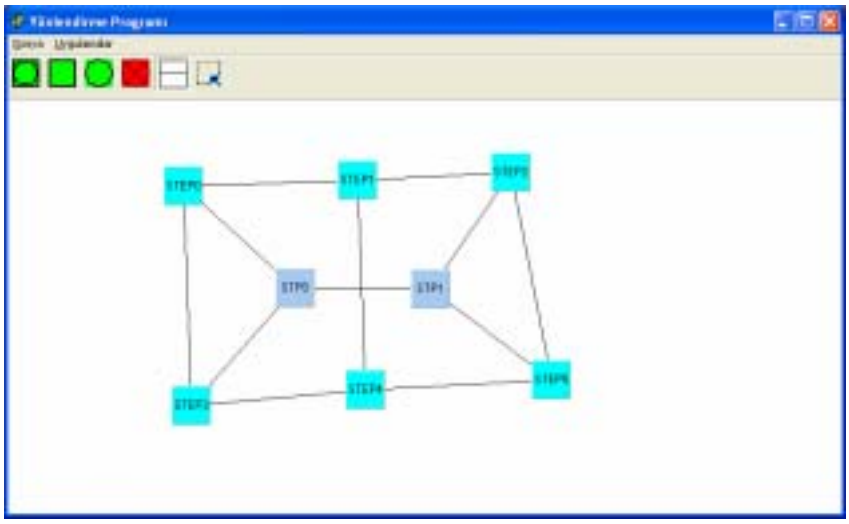


Şekil 5.23. Uygulama 2 için trafik parametrelerinin değiştirilmesi sonucu yük dağılımındaki değişiklikler

Ağ planlayıcı bu yapıda bir ağ oluşturduğunda SEP0 düğümü üzerindeki ‘toplam çağrı sayısı’ trafik parametresini 500 olarak girmek istiyorsa SEP0 – STEP0 arasındaki Link0 ve SEP0 – SEP3 arasındaki Link4 kapasitelerini 30 dan daha büyük planlamak zorundadır.

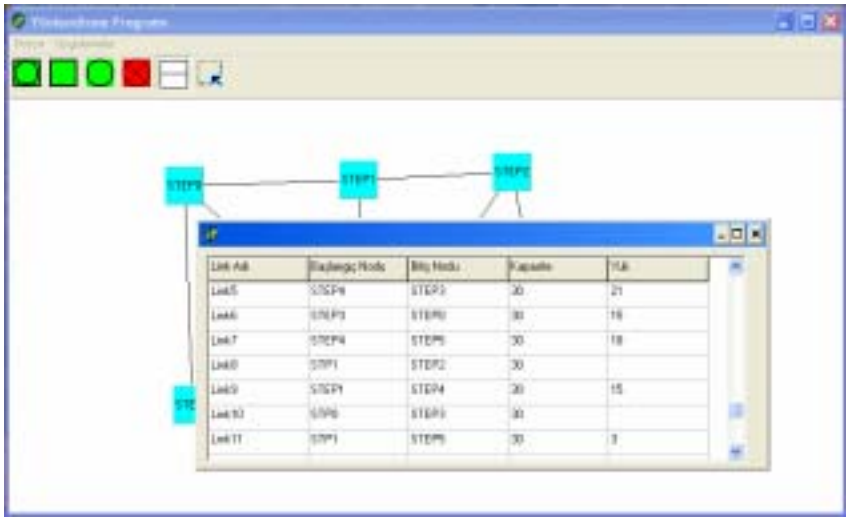
5.5.3. Uygulama 3

Bu uygulama için 6 adet STEP tipi düğüm ve 2 adet STP tipi düğüm birbirine bağlanarak Şekil 5.24’de görülen ağ oluşturulsun. Oluşturulan bu ağ için yük dağılımı hesaplanarak yük dağılım tablosu oluşturulsun.



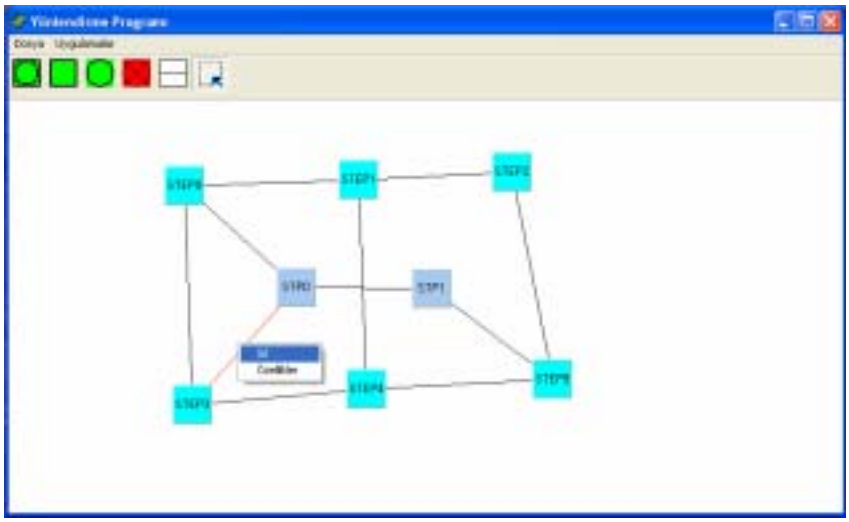
Şekil 5.24. Uygulama 3 için ağ yapısı

Şekil 5.25'den görüldüğü gibi STP1 – STEP2 düğümlerini birbirine bağlayan Link8 ve STP0 – STEP3 arasındaki Link10 üzerinden trafik geçmemektedir. Oluşturulan yönlendirme tablolarına göre bu yapıdaki bir ağ için bu bağlantılar gereksizdir ve planlayıcı tarafından silinmelidir.



Şekil 5.25. Uygulama 3 için yük dağılım tablosu

Üzerinden trafik geçmeyen linkler silinerek ağın son durumu oluşturulur. Planlayıcı bu sayede gereksiz yapılacak bir bağlantıyı dolayısıyla kaynak israfını planlama aşamasında engellemiş olacaktır.



Şekil 5.26. Uygulama 3 için üzerinden trafik geçmeyen linklerin silinmesi

6. SONUÇ VE ÖNERİLER

Bu çalışma için geliştirilen program yazılırken Delphi yazılım dili kodları kullanılmıştır. Oluşturulan düğüm santralleri, bağlantı linkleri ve yönlendirme tabloları için paradox tipi tablolardan oluşan bir veri tabanı kullanılmıştır. Yönlendirme tabloları oluşturulurken döngü kontrolü ve RIP için kullanılan algoritma, programın kurulu olduğu bilgisayarın işlemci kapasitesinin büyük bir kısmını kullanmaktadır. Bu nedenle yüksek hızlı işlemciye sahip bir bilgisayar kullanılmalıdır.

Programda kullanımı oldukça kolay kullanılabilecek bir arayüz geliştirilmiştir. Ağı oluşturacak olan düğümlerin ve linklerin oluşturulması, silinmesi veya yer değiştirilmesi kolaylıkla gerçekleştirilebilmektedir. Ancak program bir ağın nasıl oluşturulması gerektiği, yön tablolarının veya trafik yükünün ne anlama geldiği ile ilgili kullanıcıya herhangi bir bilgi vermemektedir. Bu nedenle kullanıcının haberleşme ağları ve kullanılan cihazlarla ilgili ön bilgiye sahip olması gerekmektedir.

Yönlendirme tablolarının oluşturulması ve trafik yüklerinin dağılımının hesaplanması da geliştirilen arayüzle kolaylıkla yapılmaktadır. Oluşturulan tabloların bir operatör tarafından gerçek cihazlar üzerinde tanımlanması gerekmektedir. Bu anlamda, programın ağı oluşturan düğüm santralleri ile bağlantılı hale getirilip, yön tablolarının program tarafından otomatik olarak cihazlara yüklenmesi mümkündür. Ancak bu farklı ve daha kapsamlı bir çalışma konusu olabilir.

KAYNAKLAR

1. Dimitri, P. B. “Dynamic Behavior of Shortest Path Routing Algorithms for Communication Networks”. ***IEEE Transactions on Automatic Control***, 14(2): 60–74, (1982).

2. Italtel, “Yönlendirme”, Anahtarlama Sistemi Eğitim Dökümanı Versiyon 9.85 COM-UTUT-9.85-D-35-E, ***Italtel SIS/DAR-ADC/CT***, Milan, 1-7, 26-30, 54-59 (1998).

3. ITU-T, “Telephone Network And ISDN Operation, Numbering, Routing And Mobile Service”, ITU-T Recommendation E.172, ***ITU***, 1 – 3, (1992).

4. GANS, N., KOOLE, G., MANDELBAUM, A., “Telephone Call Centers: Tutorial, Review, and Research Prospects”, ***Manufacturing & Service Operations Management***, 5(2): 113-121 (2003).

5. ITU-T, “Telephone Network And ISDN Operatıon, Numbering, Routing and Mobile Service - International Telephone Routing Plan ”, ITU-T Recommendation E.171, ***ITU***, 1 – 10, (1993).

6. Italtel, “UT Sistemine Giriş”, Anahtarlama Sistemi Eğitim Dökümanı Versiyon 9.85 COM-UTUT-9.85-D-01-E, ***Italtel SIS/DAR-ADC/CT***, Milan, 1-35 (1998).

7. Italtel, “Yönlendirme”, İşletme El Kitabı Versiyon 9.85 COM-UTF3-9.85-E- 01-E Edition E07, ***Italtel SIS/DAR-ADC/CT***, Milan, 1-7, 75-89, 189-206 (1998).

8. ITU-T, Terms and Definitions of Traffic Engineering, ITU-T Recommendation E.600, ***ITU***, 1-4 (1993).

9. Siemens, “Hat Network Design”, EWSD Switching Network Planning A30181-X1862-XA07-01-7635, ***Siemens AG***, München, 4–30, 38–61 (1999).

10. ITU, “Telephone Network and ISDN Operation, Numbering, Routing And Mobile Service - Traffic Routing”, ITU-T Recommendation E.170, ***ITU***, Geneva, 1–6, (1992).

EKLER

EK-1 Uygulama programı için kaynak kodları

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, ExtCtrls, Menus, ComCtrls, Grids, DBGrids, DB,
DBTables,Excel_TLB,ToolWin, ImgList,strutils,math;
type
TForm1 = class(TForm)
PMSwitch: TPopupMenu;
delete1: TMenuItem;
PMSigLink: TPopupMenu;
delete: TMenuItem;
TSelect: TTable;
ImageList1: TImageList;
MainMenu1: TMainMenu;
File1: TMenuItem;
RoutingSchemes1: TMenuItem;
Uygulamalar1: TMenuItem;
OtomatikYnlendirme1: TMenuItem;
im1: TImage;
ToolBar1: TToolBar;
Step: TToolButton;
Stp: TToolButton;
Sep: TToolButton;
Line: TToolButton;
Select: TToolButton;
Button1: TButton;
Simulation1: TMenuItem;
SimBox: TToolButton;
YkHesaplama1: TMenuItem;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
zellikler1: TMenuItem;
procedure delete1Click(Sender: TObject);
procedure LineClick(Sender: TObject);
procedure deleteClick(Sender: TObject);
procedure TSelectAfterOpen(DataSet: TDataSet);
procedure im1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure im1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure im1DragOver(Sender, Source: TObject; X, Y: Integer;
State: TDragState; var Accept: Boolean);
procedure OtomatikYnlendirme1Click(Sender: TObject);
procedure RoutingSchemes1Click(Sender: TObject);
procedure Simulation1Click(Sender: TObject);
procedure LoadCheck1Click(Sender: TObject);
procedure x1Click(Sender: TObject);
procedure YkHesaplama1Click(Sender: TObject);
procedure zellikler1Click(Sender: TObject);
procedure File1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
type
TSwitch = class (TPanel)
procedure MyMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure MyMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
procedure MyClick (Sender: TObject);
procedure MyEndDrag (Sender, Target: TObject; X, Y: Integer);
procedure MyStartDrag(Sender: TObject; var DragObject: TDragObject);
end;
var
Form1: TForm1;
myswitch: TSwitch;
mystrings:TStrings;
temp1,colortodel,Incolor,cnt,origX,origY,destX,destY:integer;
Totnum,Stepnum,Sepnum,Stpnum,Simnum:integer;
origXtoDel,origYtoDel,destXtoDel,destYtoDel:integer;
SimFix1OrigX,SimFix1OrigY,SimFix1DestX,SimFix1DestY:integer;
SimFix2OrigX,SimFix2OrigY,SimFix2DestX,SimFix2DestY:integer;
SimFixorig,Simfixdest,Simorigin,SimDest,CompName,CompName1,CompNa
me2:string;
brk:boolean;
//**** routes variables
TempRouteArray:array[-1..100] of string;
TempArray:array[0..1000] of string;
buffer:array[0..100] of string;
temp,Route1,Route2,CurrentNode,NextNode:string;
bufindex,bufsize,TempRouteArrayBS,TempArrayBS,index:integer;
writeflg:boolean;
implementation
uses Unit2, Unit3,unit4, Unit5;
{$R *.dfm}
procedure FindNextNode(currentnode:string;index:integer);
var
i:integer;
begin
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
for i:=0 to TempArrayBS do begin
if temparray[i]='F'+currentnode then begin
nextnode:=temparray[i+index];
break;
end;
end;
end;
procedure WriteTempArray;
begin
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('Select * from Switch.db ');
DMSigLink.QSwitch.ExecSQL;
DMSigLink.QSwitch.open;
DMSigLink.QSwitch.First;
while not DMSigLink.QSwitch.Eof do begin
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add('Select * from Siglink.db where
(ASwName=:kod1) or (BSwName=:kod1)');

DMSigLink.QSigLink.Params[0].asstring:=DMSigLink.QSwitch.Fields[0].AsStr
ing;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.open;
temparray[TempArrayBS]:='F'+DMSigLink.QSwitch.Fields[0].AsString;
inc(TempArrayBS);
while not DMSigLink.QSigLink.Eof do begin
if
DMSigLink.QSigLink.Fields[1].AsString=DMSigLink.QSwitch.Fields[0].AsStri
ng then temparray[TempArrayBS]:=DMSigLink.QSigLink.Fields[4].AsString;
if
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QSigLink.Fields[4].AsString=DMSigLink.QSwitch.Fields[0].AsStri
ng then temparray[TempArrayBS]:=DMSigLink.QSigLink.Fields[1].AsString;
inc(TempArrayBS);
DMSigLink.QSigLink.Next;
end;
temparray[TempArrayBS]:='';
inc(TempArrayBS);
DMSigLink.QSigLink.Close;
DMSigLink.QSwitch.Next;
end;
DMSigLink.QSwitch.Close;
dec(TempArrayBS);
end;

function ExistNextNode(NextNode:string):Boolean;
var i:integer;
begin
for i:=0 to TempRouteArrayBS do begin
if NextNode=TempRouteArray[i] then begin
result:=true;
break;
end
else begin
result:=false;
end;

end;
end;
function ExistArray (NextNode:string):Boolean;
var i:integer;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
begin
TempRouteArray[TempRouteArrayBS+1]:=NextNode;
DmSigLink.TblRoutes.Open;
DmSigLink.TblRoutes.First;
while not DmSigLink.TblRoutes.Eof do begin//4
for i:=0 to TempRouteArrayBS+1 do begin//3
if TempRouteArray[i]=DmSigLink.TblRoutes.Fields[i].Value then begin//1
result:=true;
end//1
else begin//2
result:=false;
break;
end;//2
end;//3
if result=true then break;
DmSigLink.TblRoutes.Next;
end;//4
TempRouteArray[TempRouteArrayBS+1]:='';
DmSigLink.TblRoutes.Close;
end;
procedure WriteArrayToDatabase;
var i:integer;
begin
DmSigLink.TblRoutes.open;
DmSigLink.TblRoutes.Append;
for i:=0 to TempRouteArrayBS do begin
DmSigLink.TblRoutes.fields[i].Value:=TempRouteArray[i];
end;
DmSigLink.TblRoutes.Refresh;
DmSigLink.TblRoutes.Close;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;
procedure FindRoute (origin,destination:string);
var index,cnt:integer;
begin
index:=20;cnt:=1;
DMSigLink.Qroutes.Open;
DmSiglink.Qroutes.First;
while not DmSiglink.Qroutes.Eof do begin//1
while  DmSiglink.Qroutes.Fields[cnt].Value<>null do begin//2
if  DmSiglink.Qroutes.Fields[cnt].Value=destination then break;
inc(cnt);
end;//2
if cnt<index then begin//3
index:=cnt;
route1:=DmSiglink.Qroutes.Fields[1].Value;
end;//3
if index=1 then break;
cnt:=1;
DmSiglink.Qroutes.Next;
end;//1
end;
procedure CreateRoutingTables;
var
i:integer;
begin//4
DMSigLink.QRouting.Open;
DMSigLink.QRouting.Append;
for i:=0 to bufsize do begin//2
if buffer[i]<>temp then begin//1
DMSigLink.QRouting.Fields[0].Value:=temp;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QRouting.Fields[1].Value:=buffer[i];
FindRoute(temp,buffer[i]);
DMSigLink.QRouting.Fields[2].Value:=route1;
DMSigLink.QRouting.Append;
end;//1
end;//2
end;//4
procedure RefreshLine;
begin
DmSiglink.TblSigLink.First;
while not dmsiglink.TblSigLink.Eof do begin
form1.im1.Canvas.Pen.Color:=DmSiglink.TblSigLink.Fields[0].Value;
form1.im1.Canvas.MoveTo(DmSiglink.TblSigLink.Fields[2].Value,DmSiglink.
TblSigLink.Fields[3].Value);
form1.im1.Canvas.LineTo(DmSiglink.TblSigLink.Fields[5].Value,DmSiglink.T
blSigLink.Fields[6].Value);
form1.im1.canvas.MoveTo(origx,origy);
DmSiglink.TblSigLink.Next;
end;
end;
procedure TSwitch.MyStartDrag (Sender: TObject; var DragObject:
TDragObject);
begin
DMSigLink.QSLDrag.Close;
DMSigLink.QSLDrag.SQL.clear;
DMSigLink.QSLDrag.sql.add('Select * from Siglink.db where
(ASwName=:kod1) or (BSwName=:kod1)');
DMSigLink.QSLDrag.Params[0].asstring:=compname;
DMSigLink.QSLDrag.ExecSQL;
DMSigLink.QSLDrag.Open;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;
procedure TSwitch.MyEndDrag (Sender, Target: TObject; X, Y: Integer);
begin
dragmode:=dmmanual;
end;
procedure TSwitch.MyMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
compname:=name;
if form1.Select.down then begin
color:=clwhite;
dragmode:=dmautomatic;
end;
form1.TSelect.Open;
end;
procedure TSwitch.MyMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
end;
procedure TSwitch.MyClick (Sender:TObject);
begin
if form1.Line.Down then begin//4
inc(cnt);
if (cnt=1) then begin//3
compname1:=name;
if compname1='SBox0' then begin simfix1origX:=left+20;simfix1origy:=top-27
end;
if compname1='SBox1' then begin simfix2origX:=left+20;simfix2origy:=top-27
end;
origx:=left+20;origy:=top-27;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;//3
if (cnt=2) then begin//2
compname2:=name;
if ansicontainsstr(compname1,'SBox') and
ansicontainsstr(compname2,'STP') then begin
showmessage('STP tipi düğüm için simülasyon bağlantısı yapılamaz..!');
form1.im1.canvas.Pen.Color:=clwhite;form1.im1.canvas.LineTo(destx,desty);
form1.im1.canvas.MoveTo(origx,origy);
cnt:=0;
end
else begin//5
form1.im1.canvas.Pen.Color:=clwhite;form1.im1.canvas.LineTo(destx,desty);
form1.im1.canvas.MoveTo(origx,origy);
destx:=left+20;desty:=top-27;
form1.im1.canvas.Pen.Color:=lncolor;form1.im1.canvas.LineTo(destx,desty);
inc(lncolor);
cnt:=0;
if ansicontainsstr(compname1,'SBox') then begin
if compname1='SBox0' then begin
SimFixOrig:=compname2;simfix1destx:=left+20;simfix1desty:=top-27;end;
if compname1='SBox1' then begin
SimFixDest:=compname2;simfix2destx:=left+20;simfix2desty:=top-27;end;
end
else begin
DmSigLink.TblSigLink.open;
with DmSigLink.TblSigLink do begin//1
Append;
fields[0].Value:=form1.im1.canvas.Pen.Color;
fields[1].Value:=compname1;
fields[2].Value:=origX;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
fields[3].Value:=origY;
fields[4].Value:=compname2;
fields[5].Value:=destX;
fields[6].Value:=destY;
fields[7].Value:=form1.im1.canvas.Pen.Color;
fields[9].Value:='Link'+fields[0].AsString;
fields[10].Value:=30;
end;//1
DmSigLink.TblSigLink.Refresh;
end;//6
end;//5
end;//2
end;//4
end;//begin
procedure TForm1.delete1Click(Sender: TObject);
begin
if ansicontainsstr(compname,'SBox') then begin
DMSigLink.TblSigLink.Open;
DMSigLink.TblSigLink.First;
while not DMSigLink.TblSigLink.Eof do begin
DMSigLink.TblSigLink.Edit;
DMSigLink.TblSigLink.Fields[0].asinteger:=DMSigLink.TblSigLink.Fields[7].a
sinteger;
DMSigLink.TblSigLink.Next;
end;
DMSigLink.TblSigLink.Refresh;
refreshline;
TSwitch(findcomponent(compname)).Destroy;
im1.Canvas.Pen.Color:=clwhite;
if compname='SBox0' then begin
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
im1.Canvas.MoveTo(simfix1origx,simfix1origy);
im1.Canvas.LineTo(simfix1destx,simfix1desty);
end;
if compname='SBox1' then begin
im1.Canvas.MoveTo(simfix2origx,simfix2origy);
im1.Canvas.LineTo(simfix2destx,simfix2desty);
end;
end
else begin
TSwitch(findcomponent(compname)).Destroy;
DMSigLink.Qswitch.Close;
DMSigLink.Qswitch.SQL.clear;
DMSigLink.Qswitch.sql.add(' delete from Switch.db where (Name=:kod1)');
DMSigLink.Qswitch.Params[0].asstring:=compname;
DMSigLink.Qswitch.ExecSQL;
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add('Select * from Siglink.db where
(ASwName=:kod1) or (BSwName=:kod1)');
DMSigLink.QSigLink.Params[0].asstring:=compname;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.open;
DMSigLink.QSigLink.First;
form1.im1.canvas.Pen.Color:=clwhite;
while not DMSigLink.QSigLink.Eof do begin
form1.im1.canvas.MoveTo(DMSigLink.QSigLink.Fields[2].value,DMSigLink.Q
SigLink.Fields[3].value);
form1.im1.canvas.LineTo(DMSigLink.QSigLink.Fields[5].value,DMSigLink.Q
SigLink.Fields[6].value);
DMSigLink.QSigLink.Next;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' delete from Siglink.db where
(ASwName=:kod1) or (BSwName=:kod1)');
DMSigLink.QSigLink.Params[0].asstring:=compname;
DMSigLink.QSigLink.ExecSQL;
end;
end;
procedure TForm1.LineClick(Sender: TObject);
begin
cnt:=0;
end;
procedure TForm1.deleteClick(Sender: TObject);
begin
im1.canvas.Pen.color:=clwhite;
im1.canvas.MoveTo(origxtodel,origytodel);
im1.canvas.LineTo(destxtodel,destytodel);
value;
Tswitch(findcomponent(TSelect.Fields[0].asstring)).dragmode:=dmmanual;
end;
TSelect.Next;
end;
TSelect.Close;
refreshline;
end;
procedure TForm1.im1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var
i,j:integer;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
begin
if (button=mbleft) and (Step.Down) then begin
myswitch:=TSwitch.Create(self);
with myswitch do begin
parent:=self;
left:=x-20;
top:=y+27;
name:='STEP'+inttostr(stepnum);inc(stepnum);
width:=40;
height:=40;
onmousedown:=mymousedown;
onmouseup:=mymouseup;
onenddrag:=myenddrag;
onclick:=myclick;
onstartdrag:=mystartdrag;
popupmenu:=pmswitch;
color:=clAqua;
end;//with myswitch do end
inc(totnum);
DmSigLink.TblSwitch.open;
with DmSigLink.TblSwitch do begin
Append;
fields[0].Value:=myswitch.name;
fields[1].Value:='STEP';
fields[2].Value:=myswitch.Color;
fields[3].Value:=100;
fields[4].Value:=5000;
DmSigLink.TblSwitch.open;
fields[6].Value:=900;
end;//with DmSigLink.TblSwitch do end
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DmSigLink.TblSwitch.Refresh;
DmSigLink.TblSwitch.First;
while not DmSigLink.TblSwitch.Eof do begin
if DmSigLink.TblSwitch.Fields[1].AsString<>'STP' then begin
DmSigLink.TblSwitch.Edit;
DmSigLink.TblSwitch.Fields[5].Value:=1/totnum;
DmSigLink.TblSwitch.Refresh;
end;
DmSigLink.TblSwitch.Next;
end;
DmSigLink.TblSwitch.Close;
//*******   Creation of STP points     ************
if (button=mbleft) and (Stp.Down) then begin
myswitch:=TSwitch.Create(self);
with myswitch do begin
parent:=self;
left:=x-20;
top:=y+27;
name:='STP'+inttostr(stpnum);inc(stpnum);
width:=40;
height:=40;
onmousedown:=mymousedown;
onmouseup:=mymouseup;
onenddrag:=myenddrag;
onclick:=myclick;
onstartdrag:=mystartdrag;
popupmenu:=pmswitch;
color:=clskyblue;
end;//with myswitch do end
DmSigLink.TblSwitch.open;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
with DmSigLink.TblSwitch do begin
Append;
fields[0].Value:=myswitch.name;
fields[1].Value:='STP';
fields[2].Value:=myswitch.Color;
end;//with DmSigLink.TblSwitch do end
DmSigLink.TblSwitch.Refresh;
end;
DmSigLink.TblSwitch.Close;
//*******   Creation of SEP points     ************
if (button=mbleft) and (Sep.Down) then begin
myswitch:=TSwitch.Create(self);
with myswitch do begin
parent:=self;
left:=x-20;
top:=y+27;
name:='SEP'+inttostr(sepnum);inc(sepnum);
width:=40;
height:=40;
onmousedown:=mymousedown;
onmouseup:=mymouseup;
onenddrag:=myenddrag;
onclick:=myclick;
onstartdrag:=mystartdrag;
popupmenu:=pmswitch;
color:=14780018;
end;//with myswitch do end
inc(totnum);
DmSigLink.TblSwitch.open;
with DmSigLink.TblSwitch do begin
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
Append;
fields[0].Value:=myswitch.name;
fields[1].Value:='SEP';
fields[2].Value:=myswitch.Color;
fields[3].Value:=100;
fields[4].Value:=5000;
fields[6].Value:=900;
end;//with DmSigLink.TblSwitch do end
DmSigLink.TblSwitch.Refresh;
end;
DmSigLink.TblSwitch.Close;
//-----
DmSigLink.TblSwitch.open;
DmSigLink.TblSwitch.First;
while not DmSigLink.TblSwitch.Eof do begin
if DmSigLink.TblSwitch.Fields[1].AsString<>'STP' then begin
DmSigLink.TblSwitch.Edit;
DmSigLink.TblSwitch.Fields[5].Value:=1/totnum;
DmSigLink.TblSwitch.Refresh;
end;
DmSigLink.TblSwitch.Next;
end;
DmSigLink.TblSwitch.Close;
//********  Creation of Simulation box  **********************************
if (button=mbleft) and (SimBox.Down) then begin
if simnum<2 then begin
myswitch:=TSwitch.Create(self);
with myswitch do begin
parent:=self;
left:=x-20;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
top:=y+27;
if simnum=0 then begin name:='SBox0';inc(simnum);end
else begin name:='SBox1';inc(simnum);end;
width:=32;
height:=32;
onmousedown:=mymousedown;
onmouseup:=mymouseup;
onenddrag:=myenddrag;
onclick:=myclick;
onstartdrag:=mystartdrag;
popupmenu:=pmswitch;
color:=clred;
end;//with myswitch do end
end;
end;
//*********************************************

DMSigLink.TblSwitch.Open;
while not DMSigLink.TblSwitch.Eof do begin
Tswitch(findcomponent(DMSigLink.TblSwitch.Fields[0].asstring)).color:=DMS
igLink.TblSwitch.Fields[2].Value;
Tswitch(findcomponent(DMSigLink.TblSwitch.Fields[0].asstring)).dragmode:=
dmmanual;
DMSigLink.TblSwitch.Next;
end;
DMSigLink.TblSwitch.Close;
if Select.Down then begin
im1.Canvas.Pen.Color:=clblack;
im1.Canvas.MoveTo(simfix1origx,simfix1origy);
im1.Canvas.LineTo(simfix1destx,simfix1desty);
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
im1.Canvas.MoveTo(simfix2origx,simfix2origy);
im1.Canvas.LineTo(simfix2destx,simfix2desty);
refreshline;
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add('Select * from Siglink.db ');
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.open;
DMSigLink.QSigLink.First;
while not DMSigLink.QSigLink.Eof do begin
for i:=-5 to 5 do begin//for i:=-5 to 5 do begin
for j:=-5 to 5 do begin//for j:=-5 to 5 do begin
if DMSigLink.QSigLink.Fields[0].Value = form1.im1.canvas.Pixels[x+i,y+j]
then begin
colortodel:=DMSigLink.QSigLink.Fields[0].Value;
origxtodel:=DMSigLink.QSigLink.Fields[2].Value;
origytodel:=DMSigLink.QSigLink.Fields[3].Value;
destxtodel:=DMSigLink.QSigLink.Fields[5].Value;
destytodel:=DMSigLink.QSigLink.Fields[6].Value;
refreshline;
form1.im1.canvas.Pen.color:=clred;
form1.im1.canvas.MoveTo(origxtodel,origytodel);
form1.im1.canvas.LineTo(destxtodel,destytodel);
if (button=mbright) then pmsiglink.AutoPopup:=true;
end;//if DMSigLink.QSigLink.Fields[0].Value =
form1.im1.canvas.Pixels[x+i,y+j] then begin
end;//for j:=-5 to 5 do end
end;//for i:=-5 to 5 do end
DMSigLink.QSigLink.Next;
end;//while not DMSigLink.QSigLink.Eof do end
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;//if Select.Down then end;
end;
procedure TForm1.im1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
if (cnt=1) and (Line.Down) then begin
refreshline;
im1.canvas.Pen.Color:=clwhite;im1.canvas.LineTo(destx,desty);im1.canvas.
MoveTo(origx,origy);
im1.canvas.Pen.Color:=lncolor;im1.canvas.LineTo(x,y);im1.canvas.MoveTo(
origx,origy);
destx:=x;desty:=y;
end;
end;
procedure TForm1.im1DragOver(Sender, Source: TObject; X, Y: Integer;
State: TDragState; var Accept: Boolean);
begin
with Sender as TImage do
begin
(Source as TSwitch).left:=x-20;
(Source as TSwitch).top:=y+27;
end;
DMSigLink.QSLDrag.First;
while not DMSigLink.QSLDrag.Eof do begin
if DMSigLink.QSLDrag.Fields[1].value = compname then begin
im1.canvas.Pen.Color:=clwhite;
im1.canvas.MoveTo(DmSiglink.QSLDrag.Fields[2].Value,DmSiglink.QSLDra
g.Fields[3].Value);
im1.canvas.LineTo(DmSiglink.QSLDrag.Fields[5].Value,DmSiglink.QSLDrag.
Fields[6].Value);
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QSLDrag.Edit;
DMSigLink.QSLDrag.Fields[2].value:=x;
DMSigLink.QSLDrag.Fields[3].value:=y;
im1.canvas.Pen.Color:=DmSiglink.QSLDrag.Fields[0].Value;
im1.canvas.MoveTo(DmSiglink.QSLDrag.Fields[2].Value,DmSiglink.QSLDra
g.Fields[3].Value);
im1.canvas.LineTo(DmSiglink.QSLDrag.Fields[5].Value,DmSiglink.QSLDrag.
Fields[6].Value);
end;
if DMSigLink.QSLDrag.Fields[4].value = compname then begin
im1.canvas.Pen.Color:=clwhite;
im1.canvas.MoveTo(DmSiglink.QSLDrag.Fields[2].Value,DmSiglink.QSLDra
g.Fields[3].Value);
im1.canvas.LineTo(DmSiglink.QSLDrag.Fields[5].Value,DmSiglink.QSLDrag.
Fields[6].Value);
DMSigLink.QSLDrag.Edit;
DMSigLink.QSLDrag.Fields[5].value:=x;
DMSigLink.QSLDrag.Fields[6].value:=y;
im1.canvas.Pen.Color:=DmSiglink.QSLDrag.Fields[0].Value;
im1.canvas.MoveTo(DmSiglink.QSLDrag.Fields[2].Value,DmSiglink.QSLDra
g.Fields[3].Value);
im1.canvas.LineTo(DmSiglink.QSLDrag.Fields[5].Value,DmSiglink.QSLDrag.
Fields[6].Value);
end;
DMSigLink.QSLDrag.Refresh;
DMSigLink.QSLDrag.Next;
end;
end;
procedure TForm1.OtomatikYnlendirme1Click(Sender: TObject);
var Loop,WriteFlag:boolean;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
begin
index:=1;TempArrayBS:=0;TempRouteArrayBS:=0;Loop:=true;WriteFlag:=tru
e;
bufsize:=0;bufindex:=0;
DMSigLink.QSwitch.Close;
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('delete  from Routing.db ');
DMSigLink.QSwitch.ExecSQL;
DMSigLink.TblSwitch.Open;
DMSigLink.TblSwitch.First;
while not DMSigLink.TblSwitch.Eof do begin//5
buffer[bufsize]:=DMSigLink.TblSwitch.Fields[0].Value;
inc(bufsize);
DMSigLink.TblSwitch.Next;
end;//5
dec(bufsize);
WriteTempArray;
DMSigLink.TblSwitch.Open;
DMSigLink.TblSwitch.First;
while not DMSigLink.TblSwitch.Eof do begin //1
temp:=DMSigLink.TblSwitch.Fields[0].Value;
CurrentNode:=DMSigLink.TblSwitch.Fields[0].Value;
TempRouteArray[TempRouteArrayBS]:=CurrentNode;
while loop do begin // Loop
FindNextNode(CurrentNode,index);
if NextNode='' then
begin // 5
if WriteFlag then WriteArrayToDatabase;
TempRouteArray[TempRouteArrayBS]:='';
dec(TempRouteArrayBS);
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
CurrentNode:=TempRouteArray[TempRouteArrayBS];
if CurrentNode='' then begin
WriteFlag:=true;TempRouteArrayBS:=0;index:=1;break;end
else begin WriteFlag:=false;index:=1;end;
end // 5
else
begin // 2
if ExistNextNode(NextNode) then begin inc(index);   end
else
begin //3
if ExistArray(NextNode) then begin inc(index);  end
else
begin // 4
inc(TempRouteArrayBS);
TempRouteArray[TempRouteArrayBS]:=NextNode;
WriteFlag:=true;
index:=1;
CurrentNode:=NextNode;
end; //4
end;//3
end; // 2
end; // Loop
CreateRoutingTables;
DMSigLink.QSwitch.Close;
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('delete  from Routes.db ');
DMSigLink.QSwitch.ExecSQL;
DMSigLink.TblSwitch.Next;
end; // 1
ykhesaplama1.Enabled:=true;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
Simulation1.Enabled:=true;
Showmessage('Routing Tabloları Oluşturuldu');
end;
procedure TForm1.RoutingSchemes1Click(Sender: TObject);
var index:integer;
begin
form3.Close;
index:=0;
form3.Stringgrid1.Cells[1,0]:='Route 1';
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add('select * from Switch.db where (Name=:kod1)');
DMSigLink.QRouting.Params[0].asstring:=compname;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
Form3.Edit1.Text:=DMSigLink.QRouting.Fields[0].AsString;
Form3.Edit2.Text:=DMSigLink.QRouting.Fields[3].AsString;
Form3.Edit3.Text:=DMSigLink.QRouting.Fields[4].AsString;
Form3.Edit4.Text:=DMSigLink.QRouting.Fields[5].AsString;
Form3.Edit9.Text:=DMSigLink.QRouting.Fields[6].AsString;
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add(' select * from Routing.db where
(Origins=:kod1)');
DMSigLink.QRouting.Params[0].asstring:=compname;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
DMSigLink.QRouting.First;
form3.stringgrid1.Cells[1,1]:=DMSigLink.QRouting.Fields[2].asstring;
//form3.stringgrid1.Cells[2,1]:=DMSigLink.QRouting.Fields[3].asstring;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
while not DMSigLink.QRouting.Eof do begin
form3.combobox1.Items.insert(index,DMSigLink.QRouting.Fields[1].Value);
inc(index);
DMSigLink.QRouting.Next;
end;
form3.Edit1.Text:=compname;
if ansicontainsstr(compname,'STEP') then form3.Edit5.Text:='STEP';
if ansicontainsstr(compname,'STP') then form3.Edit5.Text:='STP';
if ansicontainsstr(compname,'SEP') then form3.Edit5.Text:='SEP';
form3.combobox1.Text:=form3.combobox1.Items.Strings[0];
form3.Caption:=compname;
form3.Button1.Enabled:=false;
form3.Visible:=true;
end;
procedure TForm1.Simulation1Click(Sender: TObject);
var loop:boolean;
begin
DMSigLink.TblSigLink.Open;
DMSigLink.TblSigLink.First;
while not DMSigLink.TblSigLink.Eof do begin
DMSigLink.TblSigLink.Edit;
DMSigLink.TblSigLink.Fields[7].asinteger:=DMSigLink.TblSigLink.Fields[0].a
sinteger;
DMSigLink.TblSigLink.Next;
end;
DMSigLink.TblSigLink.Refresh;
loop:=true;
simorigin:=simfixorig;
simdest:=simfixdest;
im1.Canvas.Pen.Color:=claqua;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
im1.Canvas.MoveTo(simfix1origx,simfix1origy);
im1.Canvas.LineTo(simfix1destx,simfix1desty);
im1.Canvas.MoveTo(simfix2origx,simfix2origy);
im1.Canvas.LineTo(simfix2destx,simfix2desty);
while loop do begin //1
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add(' select * from Routing.db where
(Origins=:kod1) and (Destinations=:kod2)');
DMSigLink.QRouting.Params[0].asstring:=simorigin;
DMSigLink.QRouting.Params[1].asstring:=simdest;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
if DMSigLink.QRouting.Fields[2].Value=simdest then begin//2
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from SigLink.db where
((ASwName=:kod1) and (BSwName=:kod2)) or ((ASwName=:kod2) and
(BSwName=:kod1))');
DMSigLink.QSigLink.Params[0].asstring:=simorigin;
DMSigLink.QSigLink.Params[1].asstring:=simdest;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
DMSigLink.QSigLink.Edit;
DMSigLink.QSigLink.Fields[0].asinteger:=claqua;
DMSigLink.QSigLink.Refresh;
refreshline;
break;
end//2
else begin//3
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from SigLink.db where
((ASwName=:kod1) and (BSwName=:kod2)) or ((ASwName=:kod2) and
(BSwName=:kod1))');
DMSigLink.QSigLink.Params[0].asstring:=simorigin;
DMSigLink.QSigLink.Params[1].asstring:=DMSigLink.QRouting.Fields[2].Val
ue;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
DMSigLink.QSigLink.Edit;
DMSigLink.QSigLink.Fields[0].asinteger:=claqua;
DMSigLink.QSigLink.Refresh;
refreshline;
simorigin:=DMSigLink.QRouting.Fields[2].Value;
end;//3
end;//1
DMSigLink.TblSigLink.First;
while not DMSigLink.TblSigLink.Eof do begin
DMSigLink.TblSigLink.Edit;
DMSigLink.TblSigLink.Fields[0].asinteger:=DMSigLink.TblSigLink.Fields[7].a
sinteger;
DMSigLink.TblSigLink.Refresh;
DMSigLink.TblSigLink.Next;
end;
end;
procedure TForm1.LoadCheck1Click(Sender: TObject);
begin
Form4.Visible:=true;
Form4.StringGrid1.Cells[0,0]:='Line Name';
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
Form4.StringGrid1.Cells[1,0]:='Hat Sayısı';
Form4.StringGrid1.Cells[2,0]:='Trafik Hacmi';
Form4.StringGrid1.Cells[3,0]:='BHCA';
end;
procedure TForm1.x1Click(Sender: TObject);
begin
button1.Click;
end;
procedure TForm1.YkHesaplama1Click(Sender: TObject);
var yuk:real;
orig,dest,route:string;
begin//3
dmsiglink.tblrouting.First;
while not Dmsiglink.TblRouting.Eof do begin //2
route:=Dmsiglink.TblRouting.Fields[2].AsString;
dest:=Dmsiglink.TblRouting.Fields[1].AsString;
orig:=Dmsiglink.TblRouting.Fields[0].AsString;
if not ansicontainsstr(orig+dest,'STP') then begin
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from Switch.db where (Name=:kod1)');
DMSigLink.QSigLink.Params[0].asstring:=orig;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
yuk:=DMSigLink.QSigLink.Fields[6].Value*DMSigLink.QSigLink.Fields[3].Val
ue/DMSigLink.QSigLink.Fields[4].Value;
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from Switch.db where (Name=:kod1)');
DMSigLink.QSigLink.Params[0].asstring:=dest;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
yuk:=yuk*DMSigLink.QSigLink.Fields[5].Value;
yuk:=roundto(yuk,-1);
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from SigLink.db where
((ASwName=:kod1) and (BSwName=:kod2)) or ((ASwName=:kod2) and
(BSwName=:kod1))');
DMSigLink.QSigLink.Params[0].asstring:=orig;
DMSigLink.QSigLink.Params[1].asstring:=route;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
DMSigLink.QSigLink.Edit;
DMSigLink.QSigLink.Fields[8].Value:=DMSigLink.QSigLink.Fields[8].AsInteg
er+yuk;
DMSigLink.QSigLink.Refresh;
while dest <> route do begin //1
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add(' select * from Routing.db where
((Origins=:kod1) and (Destinations=:kod2))');
DMSigLink.QRouting.Params[0].asstring:=route;
DMSigLink.QRouting.Params[1].asstring:=dest;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
route:=Dmsiglink.QRouting.Fields[2].AsString;
dest:=Dmsiglink.QRouting.Fields[1].AsString;
orig:=Dmsiglink.QRouting.Fields[0].AsString;
DMSigLink.QSigLink.Close;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add(' select * from SigLink.db where
((ASwName=:kod1) and (BSwName=:kod2)) or ((ASwName=:kod2) and
(BSwName=:kod1))');
DMSigLink.QSigLink.Params[0].asstring:=orig;
DMSigLink.QSigLink.Params[1].asstring:=route;
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSigLink.Open;
DMSigLink.QSigLink.Edit;
DMSigLink.QSigLink.Fields[8].Value:=DMSigLink.QSigLink.Fields[8].AsInteg
er+yuk;
DMSigLink.QSigLink.Refresh;
end;//1
end;
Dmsiglink.TblRouting.Next;
end;//2
//Showmessage('Link Yükleri Hesaplandı');
Form4.show;
end;//3
procedure TForm1.zellikler1Click(Sender: TObject);
begin
form5.Close;
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add('select * from SigLink.db where (Color=:kod1)');
DMSigLink.QRouting.Params[0].asinteger:=colortodel;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
Form5.Edit1.Text:=DMSigLink.QRouting.Fields[9].AsString;
Form5.Edit2.Text:=DMSigLink.QRouting.Fields[1].AsString;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
Form5.Edit3.Text:=DMSigLink.QRouting.Fields[4].AsString;
Form5.Edit4.Text:=DMSigLink.QRouting.Fields[10].AsString;
form5.Button1.Enabled:=false;
Form5.Show;
end;
procedure TForm1.File1Click(Sender: TObject);
begin
DMSigLink.QSigLink.Close;
DMSigLink.QSigLink.SQL.clear;
DMSigLink.QSigLink.sql.add('delete  from Siglink.db ');
DMSigLink.QSigLink.ExecSQL;
DMSigLink.QSwitch.Close;
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('delete  from Switch.db ');
DMSigLink.QSwitch.ExecSQL;
DMSigLink.QSwitch.Close;
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('delete  from Routes.db ');
DMSigLink.QSwitch.ExecSQL;
DMSigLink.QSwitch.Close;
DMSigLink.QSwitch.SQL.clear;
DMSigLink.QSwitch.sql.add('delete  from Routing.db ');
DMSigLink.QSwitch.ExecSQL;
end;
end.
unit Unit3;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids, ComCtrls, ExtCtrls;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
type
TForm3 = class(TForm)
Panel1: TPanel;
PageControl1: TPageControl;
TabSheet1: TTabSheet;
TabSheet2: TTabSheet;
StringGrid1: TStringGrid;
ComboBox1: TComboBox;
Label1: TLabel;
Label3: TLabel;
Label5: TLabel;
Label6: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Label4: TLabel;
Edit9: TEdit;
Button1: TButton;
procedure ComboBox1Select(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button1Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);
private
{ Private declarations }
public
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
{ Public declarations }
end;
var
Form3: TForm3;
implementation
uses Unit2,unit1;
{$R *.dfm}
procedure TForm3.ComboBox1Select(Sender: TObject);
begin
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add(' select * from Routing.db where
(Origins=:kod1) and (Destinations=:kod2)' );
DMSigLink.QRouting.Params[0].asstring:=compname;
DMSigLink.QRouting.Params[1].asstring:=combobox1.Text;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
stringgrid1.Cells[1,1]:=DMSigLink.QRouting.Fields[2].asstring;
stringgrid1.Cells[2,1]:=DMSigLink.QRouting.Fields[3].asstring;
end;
procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
combobox1.Items.Clear;
end;
procedure TForm3.Button1Click(Sender: TObject);
begin
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add('select * from Switch.db where (Name=:kod1)');
DMSigLink.QRouting.Params[0].asstring:=compname;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
DMSigLink.QRouting.Edit;
DMSigLink.QRouting.Fields[0].AsString:=Form3.Edit1.Text;
DMSigLink.QRouting.Fields[3].AsString:=Form3.Edit2.Text;
DMSigLink.QRouting.Fields[4].AsString:=Form3.Edit3.Text;
DMSigLink.QRouting.Fields[5].AsString:=Form3.Edit4.Text;
DMSigLink.QRouting.Fields[6].AsString:=Form3.Edit9.Text;
DMSigLink.QRouting.Refresh;
button1.Enabled:=false;
end;
procedure TForm3.Edit2Change(Sender: TObject);
begin
button1.Enabled:=true;
end;
unit Unit4;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids;

type
TForm4 = class(TForm)
StringGrid1: TStringGrid;
procedure FormShow(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
var
Form4: TForm4;
implementation
uses Unit2;
{$R *.dfm}
procedure TForm4.FormShow(Sender: TObject);
var i:integer;
begin
i:=1;
StringGrid1.Cells[0,0]:='Link Adı';
StringGrid1.Cells[1,0]:='Başlangıç Nodu';
StringGrid1.Cells[2,0]:='Bitiş Nodu';
StringGrid1.Cells[3,0]:='Kapasite';
StringGrid1.Cells[4,0]:='Yük';
DMSigLink.TblSigLink.Open;
DMSigLink.TblSigLink.First;
while not DMSigLink.TblSigLink.Eof do begin
StringGrid1.RowCount:=StringGrid1.RowCount+1;
StringGrid1.Cells[0,i]:=DMSigLink.TblSigLink.Fields[9].AsString;
StringGrid1.Cells[1,i]:=DMSigLink.TblSigLink.Fields[1].AsString;
StringGrid1.Cells[2,i]:=DMSigLink.TblSigLink.Fields[4].AsString;
StringGrid1.Cells[3,i]:=DMSigLink.TblSigLink.Fields[10].AsString;
if
DMSigLink.TblSigLink.Fields[8].Value<DMSigLink.TblSigLink.Fields[10].Valu
e then begin
StringGrid1.Cells[4,i]:=DMSigLink.TblSigLink.Fields[8].AsString;end
else begin
StringGrid1.Cells[4,i]:='<'+DMSigLink.TblSigLink.Fields[8].AsString+'>'; end;
inc(i);
DMSigLink.TblSigLink.Next;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;
end;
unit Unit5;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, ComCtrls;
type
TForm5 = class(TForm)
PageControl1: TPageControl;
TabSheet1: TTabSheet;
Label1: TLabel;
Label3: TLabel;
Label5: TLabel;
Label6: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Button1: TButton;
procedure Button1Click(Sender: TObject);
procedure Edit4Change(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
end;
var
Form5: TForm5;
implementation
uses Unit1,Unit2, Unit3;
{$R *.dfm}
procedure TForm5.Button1Click(Sender: TObject);
begin
DMSigLink.QRouting.Close;
DMSigLink.QRouting.SQL.clear;
DMSigLink.QRouting.sql.add('select * from SigLink.db where (Color=:kod1)');
DMSigLink.QRouting.Params[0].asinteger:=colortodel;
DMSigLink.QRouting.ExecSQL;
DMSigLink.QRouting.Open;
DMSigLink.QRouting.Edit;
DMSigLink.QRouting.Fields[10].AsString:=Form3.Edit4.Text;
DMSigLink.QRouting.Refresh;
button1.Enabled:=false;
end;
procedure TForm5.Edit4Change(Sender: TObject);
begin
button1.Enabled:=true;
end;
unit Unit2;
interface
uses
SysUtils, Classes, DBTables, DB,Graphics,dialogs;
type
TDmSigLink = class(TDataModule)
QSigLink: TQuery;
```

EK-1 (Devam) Uygulama programı için kaynak kodları

```
TblSigLink: TTable;
QSwitch: TQuery;
QSLDrag: TQuery;
TblSwitch: TTable;
TblRouting: TTable;
QRouting: TQuery;
TblRoutes: TTable;
QRoutes: TQuery;
private
{ Private declarations }
public
{ Public declarations }
end;
var
DmSigLink: TDmSigLink;
implementation
uses unit1;
{$R *.dfm}
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : EKE, Mehmet

Uyruğu : T.C.

Doğum tarihi ve yeri : 17.05.1978 Seydişehir

Medeni hali : Evli

Telefon : 0 (312) 402 55 70

e-mail : mehmeteke78@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet
Lisans	Ankara Üniversitesi/ Elektronik Bölümü	1999
Lise	Karaman Lisesi	1994

İş Deneyimi

Yıl	Yer	Görev
2002-	GenKur Bşkn. Lığı.	Mühendis

Yabancı Dil

İngilizce

Yayınlar

Harran Üniversitesi, Mühendislik Fakültesi GAP V. Mühendislik Kongresi 26-28 Nisan 2006 “ISDN santraller arası haberleşmede statik yönlendirme üzerine bir uygulama”

Hobiler

Masa Tenis, Satranç, Sinema, Müzik