

**ATM AĞLARINDA YÖNLENDİRME VE  
İŞARETLEŞME PROTOKOLÜ: PNPI**

İSTANBUL  
TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
DOĞA KUTUPHANESİ

**YÜKSEK LİSANS TEZİ  
Müh. Cumhur Okan ÖZOĞUL  
504980183011**

100949

100949

**Tezin Enstitüye Verildiği Tarih : 25 Eylül 2000**

**Tezin Savunulduğu Tarih : 9 Ekim 2000**

**Tez Danışmanı : Prof. Dr. A. Emre HARMANCI**  
**Diğer Jüri Üyeleri Prof. Dr. Günsel DURUSOY**  
**Y. Doç. Dr. Sema OKTUĞ**

*[Signatures of Prof. Dr. A. Emre HARMANCI, Prof. Dr. Günsel DURUSOY, and Y. Doç. Dr. Sema OKTUĞ]*

**EKİM 2000**

## ÖNSÖZ

Bu çalışmanın oluşturulmasında bana yol gösteren ve sonsuz destek veren tez danışmanım Prof. Dr. Emre HARMANCI'ya ve yardımlarını esirgemeyen Yrd. Doç. Dr. Sema OKTUĞ'a teşekkür ederim. Ayrıca bana her türlü desteği veren aileme de teşekkürü borç bilirim.

Eylül 2000

Cumhur Okan ÖZOĞUL



## İÇİNDEKİLER

|                                                    |             |
|----------------------------------------------------|-------------|
| <b>KISALTMALAR</b>                                 | <b>vii</b>  |
| <b>TABLO LİSTESİ</b>                               | <b>ix</b>   |
| <b>ŞEKİL LİSTESİ</b>                               | <b>x</b>    |
| <b>ÖZET</b>                                        | <b>xii</b>  |
| <b>SUMMARY</b>                                     | <b>xvii</b> |
| <b>1 GİRİŞ</b>                                     | <b>1</b>    |
| <b>2 PNNI YÖNLENDİRME PROTOKOLÜ</b>                | <b>7</b>    |
| 2.1 Fiziksel Ağ                                    | 7           |
| 2.2 En Düşük Hiyerarşik Seviye                     | 8           |
| 2.2.1 Peer Gruplar ve Mantıksal Düğümler           | 8           |
| 2.2.2 Mantıksal Bağlar ve Bu Bağların Kurulumu     | 10          |
| 2.2.3 Veri Tabanının Oluşturulması                 | 10          |
| 2.2.3.1 Düğüm Bilgileri                            | 11          |
| 2.2.3.2 Topoloji Durum Bilgisi                     | 11          |
| 2.2.3.3 Topoloji Bilgilerinin Değişimi             | 11          |
| 2.2.3.4 Taşkın Yöntemi                             | 12          |
| 2.2.4 Peer Grubu Lideri Kavramı                    | 13          |
| 2.3 Hiyerarşik Seviyelerdeki Gösterim              | 13          |
| 2.3.1 Mantıksal Düğüm                              | 13          |
| 2.3.2 Hiyerarşide Yukarı Yönde Bilgi Akışı         | 15          |
| 2.3.3 Hiyerarşide Aşağı Yönde Bilgi Akışı          | 15          |
| 2.3.4 Üst Bağlar                                   | 16          |
| 2.3.5 Yönlendirme Denetim Kanalları                | 17          |
| 2.3.6 Yatay Bağlar                                 | 18          |
| 2.3.7 Topoloji Bilgilerinin Birleştirilmesi        | 18          |
| 2.3.7.1 Bağ Bilgilerinin Birleştirilmesi           | 18          |
| 2.3.7.2 Düğüm Bilgilerinin Birleştirilmesi         | 18          |
| 2.4 Yönlendirme Hiyerarşisinin Tamamlanması        | 21          |
| 2.5 Adresleme Yapısı                               | 24          |
| 2.5.1 Düğüm Adresleri                              | 25          |
| 2.5.2 Uç Sistem Adresleri                          | 25          |
| 2.6 Adres Özetlenmesi ve Ulaşılabilirlik Bilgileri | 26          |
| 2.6.1 Adres Kapsamı                                | 30          |
| 2.7 Tek Düğüm Perspektifi                          | 30          |

|                                                       |           |
|-------------------------------------------------------|-----------|
| 2.8 Güzergah Seçimi                                   | 32        |
| 2.8.1 Bağlantı Kabul Denetimi                         | 33        |
| 2.8.2 Genel Bağlantı Kabul Denetimi                   | 33        |
| <b>3 PNNI İŞARETLEŞMESİ</b>                           | <b>34</b> |
| 3.1 Ortak İşaretleşme                                 | 36        |
| 3.2 Belirlenmiş Geçiş Listesi                         | 36        |
| 3.3 Crankback                                         | 37        |
| 3.4 Crankback ile Bağlantı Kuruluşuna Bir Örnek       | 37        |
| 3.5 Crankback Oluşmasının Sebepleri                   | 44        |
| 3.5.1 Ulaşılabilirlik Hataları                        | 44        |
| 3.5.2 Kaynak Hataları                                 | 44        |
| 3.5.3 DTL İşlenme Hataları                            | 45        |
| <b>4 PNNI YÖNLENDİRMESİNDE TEMEL MEKANİZMALAR</b>     | <b>47</b> |
| 4.1 Hello Protokolü                                   | 47        |
| 4.1.1 Hello Durum Makinesi                            | 47        |
| 4.1.1.1 Hello Veri Yapısı                             | 47        |
| 4.1.1.2 Hello Durumları                               | 49        |
| 4.1.1.3 Hello Durum Değişimlerine Yol Açan Olaylar    | 51        |
| 4.1.2 Helloların Gönderilmesi                         | 54        |
| 4.1.2.1 ULIA'ların Oluşturulması                      | 56        |
| 4.2 Veri Tabanı Zaman Uyumlaması                      | 57        |
| 4.2.1 Komşu Peer Veri Yapısı                          | 59        |
| 4.2.2 Komşu Peer Durumları                            | 61        |
| 4.2.3 Komşu Peer Durum Değişiklerine Yol Açan Olaylar | 62        |
| 4.2.4 Veri Tabanı Özet Paketlerinin Gönderilmesi      | 63        |
| 4.2.5 Veri Tabanı Özet Paketlerinin Alınması          | 65        |
| 4.2.6 PTSE Talep Paketlerinin Gönderilmesi            | 69        |
| 4.2.7 PTSE Talep Paketlerinin Alınması                | 69        |
| 4.3 Topoloji Tanımı ve Dağıtımı                       | 70        |
| 4.3.1 Topoloji Bilgisi                                | 70        |
| 4.3.1.1 Topoloji Durum Parametreleri                  | 70        |
| 4.3.1.2 Düşüme Ait Bilgiler                           | 74        |
| 4.3.1.3 Ulaşılabilirlik Bilgisi                       | 76        |
| 4.3.2 PTSP'ler ve PTSE'ler                            | 76        |
| 4.3.2.1 PTSP'ler                                      | 77        |
| 4.3.2.2 PTSE'ler                                      | 77        |
| 4.3.3 Taşkın Yöntemi                                  | 80        |
| 4.3.3.1 PTSP'lerin Gönderilmesi                       | 80        |
| 4.3.3.2 Bir PTSP'nin Alınması                         | 81        |
| 4.3.3.3 Peer Yeniden Gönderme Listesinin İşletilmesi  | 85        |
| 4.3.3.4 PTSE Alındılarının Gönderilmesi               | 85        |

|                                                                  |            |
|------------------------------------------------------------------|------------|
| 4.3.3.5 Alındı Paketlerinin Alınması                             | 85         |
| 4.3.3.6 Yeni PTSE'nin veya Yeni PTSE Örneğinin Başlatılması      | 85         |
| 4.3.3.7 PTSE'nin Yaşam Süresinin Tükenmesi                       | 86         |
| 4.3.3.8 PTSE'lerin Topoloji Veri Tabanından Kaldırılması         | 87         |
| 4.3.4 Topoloji Veri Tabanındaki PTSE'lerin Yaşlandırılması       | 88         |
| 4.3.4.1 PTSE'lerin Kalan Yaşam Süresinin Hesaplanması            | 88         |
| 4.3.4.2 Kendi Kendine Oluşan PTSE'lerin Yenilenmesi              | 88         |
| 4.3.5 Topoloji Bilgisinin Tetiklenmiş Güncellemeleri             | 88         |
| 4.3.5.1 PNNI Topoloji Durum Elemanları İçin Önemli Değişiklikler | 89         |
| 4.3.5.2 Önemli Değişikliklerin PTSE'ye İşlenmesi                 | 89         |
| 4.4 Hiyerarşi                                                    | 92         |
| 4.4.1 Peer Grup Lideri                                           | 92         |
| 4.4.1.1 Peer Grup Lideri Seçim Algoritması                       | 93         |
| 4.4.1.2 PGL Seçim Veri Yapısı                                    | 93         |
| 4.4.1.3 PGL Seçim Durumları                                      | 94         |
| 4.4.1.4 Komşu Peer Durum Değişikliklerine Yol Açan Olaylar       | 97         |
| 4.4.1.5 Düğüme Ait Bilgi PTSE'sinin Gönderilmesi                 | 98         |
| 4.4.1.6 Tercih edilen PGL'nin Seçilmesi                          | 99         |
| 4.4.1.7 Liderlik Önceliği                                        | 99         |
| 4.4.1.8 Dış Bağlarda Düğüme Ait Hiyerarşi Bilgilerinin Değişimi  | 101        |
| 4.4.2 Üst Bağların İlanı                                         | 101        |
| 4.4.2.1 En Düşük Seviye Düğümlerde Üst Bağların İlanı            | 101        |
| 4.4.2.2 Üst Bağlarda Ters Yön Bilgisi                            | 102        |
| 4.4.3 Topoloji Bilgilerinin Birleştirilmesi                      | 103        |
| 4.4.3.1 Bağ Birleştirilmesi                                      | 103        |
| 4.4.3.2 Basit Düğüm Temsili                                      | 106        |
| 4.4.3.3 Kompleks Düğüm Temsili                                   | 106        |
| 4.5 Güzergah Seçimi                                              | 109        |
| 4.5.1 Genel Bağlantı Kabul Denetimi                              | 112        |
| <b>5 PNNI PROTOKOLÜNÜ İYİLEŞTİRME ÇALIŞMALARI</b>                | <b>114</b> |
| 5.1 PNNI Protokolünün Başarımını Etkileyen Faktörler             | 114        |
| 5.1.1 Hiyerarşinin Etkileri                                      | 115        |
| 5.1.2 Topoloji ve Bağ Durum Metriklerinin Toplanması             | 116        |
| 5.1.2.1 Metrik Toplanması                                        | 117        |
| 5.1.3 Hiyerarşik Yönlendirme Algoritmaları                       | 121        |
| 5.1.4 QoS Garantisinin Sağlanması                                | 122        |
| 5.1.5 PNNI Protokolünün Ağa Getirdiği Ek Mesaj Yüğü              | 123        |
| 5.1.6 Crankback Mekanizması                                      | 124        |
| 5.2 PNNI Protokolünde Son Çalışmalar ve Eklentiler               | 125        |
| 5.2.1 Ağlararası Protokollerinin ATM üzerinden Yönlendirilmesi   | 125        |
| 5.2.1.1 Geliştirilmiş Yönlendirmeli PNNI                         | 127        |

|                                                                       |            |
|-----------------------------------------------------------------------|------------|
| 5.2.1.2 Birleřtirilmiř PNNI                                           | 130        |
| 5.2.2 Genel Uygulama Aktarımı                                         | 134        |
| 5.2.3 PNNI İřaretleřmesine Gvenlik Eklentileri                       | 135        |
| 5.2.4 Gzergah ve Baęlantı İzleme Eklentisi                           | 135        |
| 5.2.5 PNNI Mobillik Eklentisi                                         | 136        |
| 5.2.5.1 Mobil Aę, Mobil Anahtar, Eriřim Noktası Anahtarı ve Mobil Baę | 137        |
| 5.2.5.2 Mobil Mantıksal Grup Dęm                                   | 138        |
| 5.2.5.3 Aę Yapısı                                                     | 139        |
| 5.2.6 Yeniden Ynlendirme                                             | 140        |
| <b>6 SONULAR</b>                                                     | <b>144</b> |
| <b>KAYNAKLAR</b>                                                      | <b>147</b> |
| <b>ZGEMİř</b>                                                       | <b>150</b> |



## KISALTMALAR

|               |                                    |
|---------------|------------------------------------|
| <b>ABR</b>    | kullanılabilir bit hızı            |
| <b>AFI</b>    | yetkilendirme ve format tanıtıcısı |
| <b>AvCR</b>   | kullanılabilir hücre oranı         |
| <b>AW</b>     | yönetimsel ağırlık                 |
| <b>BGP</b>    | sınır geçit protokolü              |
| <b>CAC</b>    | bağlantı kabul denetimi            |
| <b>CBR</b>    | sabit bit hızı                     |
| <b>CDV</b>    | hücre gecikme değişimi             |
| <b>CLR</b>    | hücre kayıp oranı                  |
| <b>CRM</b>    | hücre oranı marjini                |
| <b>CTD</b>    | hücre aktarım gecikmesi            |
| <b>DSP</b>    | alana özgü bölüm                   |
| <b>DTL</b>    | belirlenmiş geçiş listesi          |
| <b>ES</b>     | uç sistem                          |
| <b>ESI</b>    | uç sistem tanıtıcısı               |
| <b>FSM</b>    | sonlu durum makinesi               |
| <b>GAT</b>    | genel uygulama aktarımı            |
| <b>GCAC</b>   | genel bağlantı kabul denetimi      |
| <b>IE</b>     | bilgi elemanları                   |
| <b>IG</b>     | bilgi grubu                        |
| <b>I-PNNI</b> | birleştirilmiş PNNI                |
| <b>IWF</b>    | ortak çalışan işlevler             |
| <b>LGN</b>    | mantıksal düğüm                    |
| <b>maxCR</b>  | maksimum hücre oranı               |
| <b>maxCTD</b> | maksimum hücre aktarım gecikmesi   |
| <b>NNI</b>    | ağdan-ağa arayüz                   |

|             |                                       |
|-------------|---------------------------------------|
| <b>PAR</b>  | geliştirilmiş yönlendirmeli PNNI      |
| <b>PG</b>   | peer grup                             |
| <b>PGL</b>  | peer grup lideri                      |
| <b>PNNI</b> | özel ağdan-ağa arayüz                 |
| <b>PTSE</b> | PNNI topoloji durum elemanı           |
| <b>PTSP</b> | PNNI topoloji durum paketleri         |
| <b>PVC</b>  | kalıcı sanal devreler                 |
| <b>PVCC</b> | kalıcı sanal kanal bağlantısı         |
| <b>PVPC</b> | kalıcı sanal güzergah bağlantısı      |
| <b>QoS</b>  | hizmet kalitesi                       |
| <b>RAIG</b> | kaynak kullanılabilirliği bilgi grubu |
| <b>RCC</b>  | yönlendirme denetim kanalı            |
| <b>RIP</b>  | yönlendirme bilgi protokolü           |
| <b>SVC</b>  | anahtarlı sanal devreler              |
| <b>SVCC</b> | anahtarlı sanal kanal bağlantısı      |
| <b>TVL</b>  | tip / uzunluk / değer                 |
| <b>UBR</b>  | belirlenmemiş bit hızı                |
| <b>ULIA</b> | üst bağ bilgi özelliği                |
| <b>UNI</b>  | kullanıcı-ağ arayüzü                  |
| <b>VBR</b>  | değişken bit hızı                     |
| <b>VCC</b>  | sanal kanal bağlantısı                |
| <b>VCI</b>  | sanal kanal tanıtcısı                 |
| <b>VP</b>   | sanal güzergah                        |
| <b>VPC</b>  | sanal güzergah bağlantısı             |
| <b>VPI</b>  | sanal güzergah tanıtcısı              |

## TABLO LİSTESİ

|                                                               | <u>Sayfa No</u> |
|---------------------------------------------------------------|-----------------|
| <b>Tablo 2.1.</b> Yapılandırılmış Özet Adresler .....         | 26              |
| <b>Tablo 2.2.</b> Özet Adres Listesi.....                     | 28              |
| <b>Tablo 2.3.</b> Tanıtılan Ulaşılabilir Adres Önekleri ..... | 28              |
| <b>Tablo 2.4.</b> Tanınan Ulaşılabilirlik Bilgisi .....       | 29              |
| <b>Tablo 4.1.</b> Topoloji Durum Parametreleri.....           | 71              |



## ŞEKİL LİSTESİ

|                                                                              | <u>Sayfa No</u> |
|------------------------------------------------------------------------------|-----------------|
| Şekil 1.1 : ATM ağlarının birbirine bağlanması.....                          | 2               |
| Şekil 1.2 : ATM Anahtar Operasyonu.....                                      | 2               |
| Şekil 1.3 : Sanal Devre ve Sanal Güzergah Anahtarlanması.....                | 3               |
| Şekil 1.4 : Bağlantı Kurulumu.....                                           | 4               |
| Şekil 1.5 : Bağlantı Tipleri .....                                           | 4               |
| Şekil 2.1 : Örnek ATM ağı .....                                              | 8               |
| Şekil 2.2 : PNNI hiyerarşisine kısmen sahip bir ATM ağı.....                 | 9               |
| Şekil 2.3 : Kısmen yapılandırılmış PNNI hiyerarşisi .....                    | 14              |
| Şekil 2.4 : Üst bağlar .....                                                 | 17              |
| Şekil 2.5 : Şekil 2.3'teki LGN A.4'ün kompleks düğüm gösterimi .....         | 19              |
| Şekil 2.6 : Hiyerarşik olarak tamamen yapılandırılmış bir ağ.....            | 21              |
| Şekil 2.7 : Üst bağlar .....                                                 | 23              |
| Şekil 2.8 : NSAP Adresleme Yapısı ve Adres Ön Ekleri.....                    | 24              |
| Şekil 2.9 : PNNI Hiyerarşisinde Adres Özetlemesi.....                        | 27              |
| Şekil 2.10 : A.3.3, A.3.2, A.3.1 ve A.3.4 düğümlerinden ağın görüntüsü ..... | 31              |
| Şekil 3.1 : ATM arayüzleri .....                                             | 34              |
| Şekil 3.2 : Örnek bir Hiyerarşik Ağ.....                                     | 38              |
| Şekil 3.3 : Crankback ve Çok Seçenekli Yönlendirmeye Örnek.....              | 38              |
| Şekil 4.1 : Hello Durum Değişiklikleri .....                                 | 50              |
| Şekil 4.2 : Komşu Peer Durum Değişiklikleri.....                             | 61              |
| Şekil 4.3 : PGL Seçim FSM'i .....                                            | 95              |
| Şekil 4.4 : Beş sınır düğümüne sahip bir peer grubu .....                    | 106             |
| Şekil 4.5 : Varsayılan Düğüm Temsili .....                                   | 107             |
| Şekil 4.6 : İstisnaları olan Kompleks Düğüm Temsili .....                    | 108             |
| Şekil 5.1 : Bağ Metrikleri .....                                             | 117             |
| Şekil 5.2 : Metrik Toplanması .....                                          | 118             |
| Şekil 5.3 : Değişik bağ özelliklerinde metrik toplanması.....                | 119             |

|                   |                                                            |     |
|-------------------|------------------------------------------------------------|-----|
| <b>Şekil 5.4</b>  | : PNNI ve OSPF'yi kullanan Ağ .....                        | 128 |
| <b>Şekil 5.5</b>  | : Sahte Düğümlü Mantıksal Yönlendirici Topolojisi .....    | 129 |
| <b>Şekil 5.6</b>  | : Varsayılan SVC'li Mantıksal Yönlendirici Topolojisi..... | 130 |
| <b>Şekil 5.7</b>  | : Beraber Çalışma Mimarisi .....                           | 135 |
| <b>Şekil 5.8</b>  | : Erişim Noktası Anahtarı Değişen Mobil Ağ.....            | 137 |
| <b>Şekil 5.9</b>  | : Bir ağın yapısı .....                                    | 139 |
| <b>Şekil 5.10</b> | :Yeniden Yönlendirme Örneği .....                          | 141 |



## ATM AĞLARINDA YÖNLENDİRME VE İŞARETLEŞME PROTOKOLÜ: PNNI

### ÖZET

ATM Forum tarafından standartlaştırılan Özel Ağdan-Ağa Arayüz (PNNI), ATM ağlarına esnek ölçeklenebilir ve QoS destekli bir yönlendirme mimarisi sağlamaktadır. PNNI protokolünün iki temel işlevi vardır:

1. PNNI, ağ topoloji bilgisini güvenilir olarak dağıtmak için bir yönlendirme protokolüne sahiptir. Bu sayede, herhangi bir hedef adrese giden güzergah belirlenebilir.
2. PNNI, noktadan-noktaya ve noktadan-çok noktaya bağlantıların kurulması ve sonuçlandırılması için bir işaretleşme protokolü içerir.

PNNI yönlendirme protokolü, anahtarlar ve anahtar kümeleri arasında topoloji bilgileri dağıtması için tanımlanmıştır. Bu bilgi, ağ üzerinde gerekli güzergah hesaplamalarını yapmak için kullanılır. Hiyerarşi mekanizması, bu protokolün geniş çaplı ATM ağlarına da ölçeklenebilirliğini sağlar. PNNI topolojisi ve yönlendirmesi bağ durumu yönlendirmesini baz alır.

PNNI yönlendirme protokolü, tüm bir ağı, peer gruplardan oluşmuş bir küme olarak görmektedir. Aynı peer grup kimliğine sahip olan anahtarlar, aynı peer gruba aittir. Aynı peer grubun bütün elemanları, grubun ortak bir görüntüsüne sahiptir. Her düğümün kendine ait bir topoloji veritabanı vardır. Düğümler, çevrelerindeki, veritabanlarında bulunan bilgilere göre algırlar.

Bir peer grup lideri (PGL), peer grubunu bir üst seviyedeki hiyerarşide temsil eder. Bir üst hiyerarşik seviyede bulunan peer grup liderleri yeni bir peer grup oluştururlar. Peer grup lideri bir üst hiyerarşik seviyede, mantıksal düğüm adını alır. Peer grup liderinin PNNI hiyerarşisinin bütünlüğünün sağlanmasında özel bir görevi vardır; bu görev bilginin toplanması ve dağıtılmasıdır.

Hiyerarşinin her seviyesinde, peer grup kendini açıklayan bir harita oluşturur. Bu haritalar, peer grup içerisinde, bağ durumu yönlendirmesinde olduğu gibi dağıtılırlar. Dış kullanım için peer grup lideri iç bilgiyi baz alan bir harita oluşturur. Peer grup lideri baba peer grup haritasını çocuk peer gruba aktarır. Bu harita, peer grubuna kendi çevresi hakkında bilgi aktarır. Bilgi, hiyerarşide yukarı yönde aktarılırken özetlenir. Aynı zamanda en üst seviyedeki topoloji bilgisi de aşağı yönde tüm düğümlere aktarılır. Bu durum daha düşük seviyelerdeki peer gruba dahil düğümlerin, PNNI yönlendirme alanı üzerinden erişilebilir tüm hedeflere ulaşabilmesini sağlar.

Topoloji bilgilerinin birleştirilmesi, geniş bir ağ içinde, düğüm bilgileri ve bağ bilgilerinin, ağın ölçeklenebilir olması amacıyla, indirgenmesi işlemidir. Kompleks düğüm temsili, baba peer grup içinde, düğüm bilgilerinin birleştirilmesi işleminin sonuçlarının gösterilmesinde kullanılır. Düğümler kendi kompleks düğüm temsillerini taşkın yöntemiyle, PNNI topoloji durum elemanlarını kullanarak dağıtırlar. Mantıksal düğümler, komşu peer gruplarından kompleks düğüm gösterimlerini aldıkları zaman, bu bilgileri temsil ettikleri peer grubunda taşkın yöntemiyle dağıtırlar. Bu işlemlerin sonunda bütün düğümler kendi baba peer grupları içindeki komşu peer gruplar hakkında kompleks bilgilere sahip olurlar.

PNNI protokolü en yüksek seviyede tek bir peer grup oluşuncaya kadar ağ içerisinde daha üst seviyedeki peer grupları oluşturmaya devam eder. En üst seviyedeki peer grubun bir peer grup liderine gereksinimi yoktur, çünkü kendisinin temsil edileceği daha üst seviyeli bir peer grup bulunmamaktadır. Bu şekilde birbirini takip eden en fazla 104 hiyerarşik seviye olabilir.

Mantıksal düğümler birbirlerine mantıksal bağlar ile bağlıdırlar. Hiyerarşinin en alt seviyesinde mantıksal bir bağ, ya fiziksel bir bağ ya da en düşük seviyeli iki düğüm arasındaki bir Sanal Güzergah Bağlantısı (VPC) olabilir. Bir Peer grubun içinde kalan mantıksal bağlara “yatay bağlar” denir; iki peer grubu birleştiren bağlara ise “dış bağlar” veya “üst bağlar” denir. Dış bağlar üzerinden herhangi bir veritabanı alışverişi yoktur. Dış bağlar aracılığıyla bağlanmış düğümler, veritabanı bilgilerini, taşkın yöntemini kullanarak baba peer grupları üzerinden birbirlerine aktarırlar. Sınır düğümleri, dış bağlar içinden kendilerine göre daha yüksek seviyedeki peer gruplarının bilgilerini ve bu peer gruplarında kendilerini temsil eden mantıksal grup

düğümlelerini katarak Hello protokolünü devam ettirirler. Bu bilgi sınır düğümlerinin diğer sınır düğümleri arasında ortak olan en düşük seviyeli peer gruplarının belirlemesini sağlar. Bir düğüm bu türde bağları üst bağ olarak tanır.

Hello protokolü ve taşkın yöntemi, topoloji veritabanlarının zamanuyumlu olması amacıyla veritabanı bilgilerinin alışverişinde kullanılırlar. Bu protokoller fiziksel bağlar üzerinden gönderilen çeşitli paketler kullanılırlar.

En düşük seviyede, bir mantıksal bağ çalışmaya başlayınca, ona bağlı olan düğümler PNNI Yönlendirme Denetim Kanalı (RCC) olarak bilinen bir Sanal Kanal Bağlantısı (VCC) üzerinden bilgileri değiş tokuş etmeye başlar. Her düğüm periyodik olarak “Hello paketleri” gönderir. Bu Hello paketlerinde, ATM adresleri, düğüm kimlikleri, peer grup kimlikleri ve bağ için kullandıkları kapı kimlikleri bulunmaktadır. Komşu düğümler peer grup kimliklerini “Hello paketleri” içinde değiş tokuş ederler. Aynı peer grup kimliğine sahip düğümler, aynı peer grubuna aittirler. Farklı peer grup kimliklerine sahip olan komşu düğümler, kendi peer gruplarının sınır düğümleridir.

Aynı peer grubunda olduklarını anlayan komşu düğümler, kendi topoloji veritabanlarını zamanuyumlu hale getirirler. Veritabanı zamanuyumlaması, komşu düğümler arasındaki bilgi alışverişidir ve komşu iki düğümün aynı topoloji veritabanına sahip olmasını amaçlar. Bir bağ, ancak ilgili komşu düğümler arasındaki veritabanının zaman uyumlu hale gelmesinden sonra, PNNI Topoloji Durum Elemanı (PTSE) yayını ile tanıtılabilir.

Hello paketleri değiş tokuş yapıldıktan sonra, her bir düğüm kendi kimliğini kapasitesini, grup lideri seçiminde ve PNNI hiyerarşisini yaratmada kullandığı verileri birleştirerek PNNI topoloji durum elemanlarını (PTSE) oluşturur. Bu PTSE’ler, peer grup içerisinde taşkın yöntemiyle güvenilir bir şekilde dağıtılırlar.

Bir düğümün topoloji veritabanı, PNNI yönlendirme alanında düğümün o anki durumunu gösteren, o ana dek alınmış PTSE’lerin bütününden oluşur. Topoloji veritabanı, bir düğümden, yönlendirme alanındaki ulaşılabilir diğer herhangi bir adrese gidebilmek için gerekli güzergah bilgilerini sağlar.

Düğüm kendi topoloji veritabanını, komşu bir düğüme, veritabanı özet paketleri (DBSP) göndererek açıklar. Komşu düğüm yeni bilgiler içeren PTSE’leri, PTSE

talep paketleri göndererek ister. PTSE'ler gönderilmeden önce PNNI topoloji durum paketleri halinde (PTSP) paketlenir. PTSE alındı paketleri, komşu düğümden gelen bilgilerin alındığını onaylamakta kullanılır. PTSE'ler, bağ durum parametreleri ve düğüm durum parametrelerini kapsayan topoloji durum parametrelerini içerir. Taşkın mekanizması, bir peer grup içerisindeki her düğümün aynı topoloji veritabanına sahip olmasını sağlar.

İşaretleşme için ikinci bir protokol kullanılır, bu ise ATM ağında noktadan-noktaya ve noktadan-çok noktaya bağlantı kurmada kullanılır. Bu protokolde ATM Forumun UNI işaretleşmesi baz alınmıştır, kaynaktan yönlendirme ve bir bağlantı hatasının ortaya çıkabileceği durumlara karşı crankback ve çok seçenekli yönlendirmeyi destekleyecek mekanizmalar da eklenmiştir.

Yönlendirmenin döngüsüz bir güzergahı takip etmesini sağlamak ve kaynağın yerel politikayı kullanabilmesine izin vermek için, bir mesaj içerisinde, işaretleşme için hiyerarşik kaynaktan yönlendirme ile belirlenen güzergahı tutan Belirlenmiş Geçiş Listesi (DTL) adı verilen yığın yapısı kullanılır.

Kaynak anahtar, elindeki ağ bilgilerini kullanarak bir geçiş listesi belirler. Bu listede hedefe ulaşmak için geçilmesi gereken en üst düzey peer gruplar da yer alır. Daha sonra her ara peer grup seviyesi için kullanılacak güzergah belirlenir. Sonuç olarak, birbirini takip eden üst seviyeli peer gruplardan geçen bir güzergah yığını elde edilir. DTL, bir peer grubu geçerken kullanılan tüm düğüm ve kapı kimliklerini de barındıran tam bir güzergahtır. Yığının en üstünde yer alan DTL ise en düşük seviyeli peer gruba ait DTL'dir.

Bazı durumlarda topoloji bilgileri ağın gerçek durumunu tam olarak yansıtmayabilir. Bunun bir sonucu olarak, düğümler geçersiz güzergahlar barındıran DTL'ler oluşturabilir. Böyle istenmeyen durumlar ise crankback mekanizması kullanılarak çözülmeye çalışılır. Hatayı algılayan düğüm çağrıyla geriye gönderir. Bu şekilde çağrı DTL'i yaratan düğüme kadar, problem ve crankback seviyesi de belirtilerek, geri gönderilir.

Bağlantı kurulması iki adımdan oluşur: Güzergah belirlenmesi ve Güzergah boyunca her noktada bağlantı durumunun ayarlanması.

Güzergah seçimi, eldeki bilgilere göre, talep edilen bant genişliği ve hizmet kalitesini destekleyebilecek şekilde gözüken güzergahın seçimiyle gerçekleştirilir. Eğer bir düğüm talep edilen hizmet kalitesini (QoS) destekleyemezse, crankback gerçekleşir ve yeni bir güzergah belirlenir.

Genel Bağlantı Kabul Denetimi (CAC), teklif edilen bağlantının, bağın kaynaklarının destekleyebileceği düzeyde olup olmadığının tahmin etmek için güzergah seçiminde kullanılır. Her anahtar sisteminin, genel CAC tarafından kullanılabilir, topoloji durum parametreleri kümesini ilan etmesi gerekir.

Bağlantı kurulumu boyunca, belirlenen güzergahta yer alan her bir anahtar sistemi, bağlantının talep ettiği QoS garantisini, varolan bağlantıların QoS garantilerini tehlikeye atmadan sağlayabileceğinden emin olmak için, CAC'ı uygular. Anahtar sistemlerinin uygulayacakları CAC işleminin belli bir standardı bulunmamaktadır.

PNNI protokolü, garantili bant genişliği ve sınırlanmış gecikme gibi gerçek zamanlı uygulamaların gereksinimlerini karşılamak amacıyla geliştirilmiştir. Bu gereksinimler, PNNI tarafından gerçekleştirilen yönlendirme ve işaretleşmede özel talepler oluştururlar.

Burada altı çizilmesi gereken, bu mekanizmaların ATM'e özel sorunları çözmek için değil, uygulamaların gereksinimleri tarafından oluşan sorunları çözmek için gerekli olduğudur. Bant genişliğini ve gecikme garantisini gerektiren uygulamalar yaygınlaştıkça, bu sorunların bütün ağ ortamlarında (internet dahil) çözülmesi gerekli olacaktır. ATM'e ek olarak IP ağlarında PNNI yönlendirmesinin kullanımının güncel ve cazip bir fikir olması, ATM Forum'u, her türlü ortamda yönlendirme için Birleştirilmiş PNNI (I-PNNI) ve Yönlendirmesi Artırılmış PNNI (PAR) geliştirmek için teşvik etmiştir.

## **ROUTING AND SIGNALLING PROTOCOL IN ATM NETWORKS: PNNI**

### **SUMMARY**

The Private Network-to-Network Interface (PNNI), standardized by the ATM Forum, provides a flexible, scalable and QoS-based routing architecture for ATM networks. The PNNI protocol has two main functions:

1. PNNI includes a routing protocol for reliably distributing network topology information so that paths can be computed to any addressed destination.
2. PNNI includes a signalling protocol for the establishment and takedown of point-to-point and point-to-multipoint connections.

The PNNI routing protocol is defined for distributing topology information between switches and clusters of switches. This information is used to compute paths through the network. A hierarchy mechanism ensures that this protocol scales well for large ATM networks. PNNI topology and routing is based on the link-state routing technique.

The PNNI routing protocol views the world as a collection of peer groups. Switches that have identical Peer Group ID belong to the same peer group. All members of the group maintain an identical view of the group. Each node has its own topology database. Nodes view the world according to the information presented in these databases.

A peer group leader (PGL) presents the peer group in the next level of hierarchy. In the next level of hierarchy several PGLs form a new peer group. The PGL is called a logical group node in the next level of hierarchy. The PGL has a specific role in aggregating and distributing of information for maintaining the PNNI hierarchy.

At each level of hierarchy, a peer group provides a map that describes itself. These maps are distributed in a link state fashion within the peer group. For external use the peer group leader creates a map based on the internal information. The peer group leader injects the parent peer group maps into the child peer group. The map describes the peer group for its containing environment. Information is summarized as it moves up the hierarchy. Also the information from the top level topology will propagate down to all individual nodes. This is necessary to allow nodes in the lower level peer groups to route to all destinations reachable via the PNNI routing domain.

Topology aggregation is the notion of reducing nodal as well as link information to achieve scaling in a large network. The complex node representation is used to represent the result of node aggregation in the parent peer group. Nodes distribute their complex node representations via flooding of PNNI Topology State Elements (PTSEs). When logical nodes receive complex node representations of their neighboring peers, they flood them across the peer group that they represent. At the end all the nodes have a complex node representation of neighboring peers in their parent's peer group.

PNNI protocol creates even higher level peer groups until the entire network is encompassed in a single highest level peer group. The highest level peer group does not need a PGL because there is no higher level peer group in which to represent it. There can be as much as 104 levels of ancestry.

Logical nodes are connected by logical links. At the lowest level of hierarchy, a logical link is either a physical link or a VPC between two lowest-level nodes. Logical links inside a peer group are "horizontal links" whereas links that connect two peer groups are "outside links" or "uplinks". There is no database exchange across outside links. Nodes that are connected by an outside link flood database information to each other through the parent peer groups. Border nodes extend the Hello protocol across outside links to include information about their respective higher level peer groups, and the logical group nodes representing them in these peer groups. This information allows the border nodes to determine the lowest level peer group common to both border nodes. A node advertises these links as uplinks.

The Hello protocol and Flooding mechanism are used in a database exchange process in order to synchronize topology databases. These protocols provide different packets to be sent over the physical links.

At the lowest level when a logical link becomes operational, the attached nodes initiate an exchange of information via a well-known VCC as a “PNNI Routing Control Channel” (RCC). Each node sends “Hello packets” periodically. They include private ATM address, node ID, peer group ID, and its port ID for the link. Neighboring nodes exchange peer group IDs in “Hello packets”. If they have the same peer group ID then they belong to the same peer group, otherwise they belong to different peer groups. Neighboring nodes with different peer group IDs are border nodes of their peer groups.

When neighboring nodes conclude that they are in the same peer group, they proceed to synchronize their topology databases. Database synchronization is the exchange of information between neighbor nodes resulting in the two nodes having identical topology databases. A link is advertised via PTSE transmission only after the database synchronization between the respective neighboring nodes is successfully completed.

After the Hello packets are exchanged each node bundles its state information and generates a PNNI Topology State Elements (PTSE) that describes its own identity and capabilities, information used to elect the peer group leader, and information used in establishing the PNNI hierarchy. PTSEs are reliably flooded throughout the peer group.

A node’s topology database consists of a collection of all PTSEs received, which represent that node’s present view of the PNNI routing domain. The topology database provides all the information required to compute a route from the given node to any address reachable in or through that routing domain.

The node describes its topology database by sending Database Summary Packets (DBSP) to the neighboring peer. The neighboring peer requires the newly discovered PTSEs by sending PTSE Request Packet. These PTSEs are packet into PTSPs (PNNI Topology State Packets) before the transmission. PTSE Acknowledgment Packets are used to acknowledge receipt of PTSEs from a neighbor node. PTSEs contain

topology state parameters that include link state parameters and nodal state parameters. Flooding mechanism insures that each node in a peer group maintains an identical topology database.

A second protocol is defined for signalling, that is message flows used to establish point-to-point and point-to-multipoint connections across the ATM network. This protocol is based on the ATM Forum UNI signalling, with mechanisms added to support source routing, crankback, and alternate routing of call setup requests in case of connection setup failure.

In order to ensure that routing follows a loop free path, and to allow source to utilize local policy, the path for a signaling is specified in the message in the form of a hierarchically complete source route known as a Designated Transit List, or DTL, stack.

The originating switch examines the full set of maps, as collected by the routing protocol, for the local peer group, and all visible higher level peer groups, including the map for the peer group which advertises reachability to the called party address. Using this, it then builds a sequence of paths. At the top, there is a path across the highest necessary level of peer groups to get to the destination from a parent that contains the source. Then, for each intermediate peer group level, a source route across that containing peer group is build. The result is a stack of source routes across successively higher levels of peer group. Each source route is known as a Designated Transit List (DTL). A DTL is a complete path across a peer group, consisting of a sequence of Node IDs and optionally Port IDs traversing the peer group. The DTL at the top of the stack is the DTL corresponding to the lowest level peer group.

In some cases the topology information may not be accurate. Then a node can build a DTL that includes broken paths. These situations cause a crankback process to take action. The node that detects the failure cranks the call back. The call will be cranked back to the switch who created the DTL with an indication of the problem and a Crankback Level.

Call establishment consists of two operations: The selection of path, and the setup of the connection state at each point along that path.

Path selection is done in such a way that the path chosen appears to be capable of supporting the Quality of Service (QoS) and bandwidth requested, based on currently available information. If a node cannot support the requested QoS then crankback occurs and a new path is chosen.

A generic Connection Admission Control (CAC) is needed in the path selection process to determine if a link is likely to have enough resources available to support the proposed connection. Each switching system is required to advertise a set of topology state parameters carrying some information that can be used by the generic CAC.

During connection setup, each switching system along the chosen path performs CAC to insure that the connection can be accommodated without jeopardizing QoS guarantees to the existing connections. How a switching system does CAC is not subject for standardization.

The PNNI protocol was developed to address the needs of applications with real-time requirements such as guaranteed bandwidth and bounded delay. These requirements place special demands on routing and signalling that are addressed by PNNI.

It is important to note that these mechanisms are not needed to solve problems specific to ATM, but rather to solve problems introduced by the requirements of the applications. When applications requiring bandwidth or delay guarantees become wide spread, it will be necessary to solve these problems in all networking environments, including the Internet. The idea that PNNI routing would be a good choice for use in IP networks in addition to ATM has motivated the ATM Forum to develop Integrated PNNI (I-PNNI) and PNNI Augmented Routing (PAR) for routing in both environments.

## 1 GİRİŞ

Asenkron Transfer Modu (ATM) mevcut LAN ve WAN teknolojileri üzerinden, ölçeklenebilir bant genişliği, yüksek performans ve hizmet kalitesi garantisi (QoS) gibi önemli avantajlar sunan bir ağ katmanı protokolüdür. Bu avantajlar, özellikle multimedya uygulamaları gibi hızla gelişen yeni hizmet sınıflarının uygulanmasını kolaylaştırmaktadır.

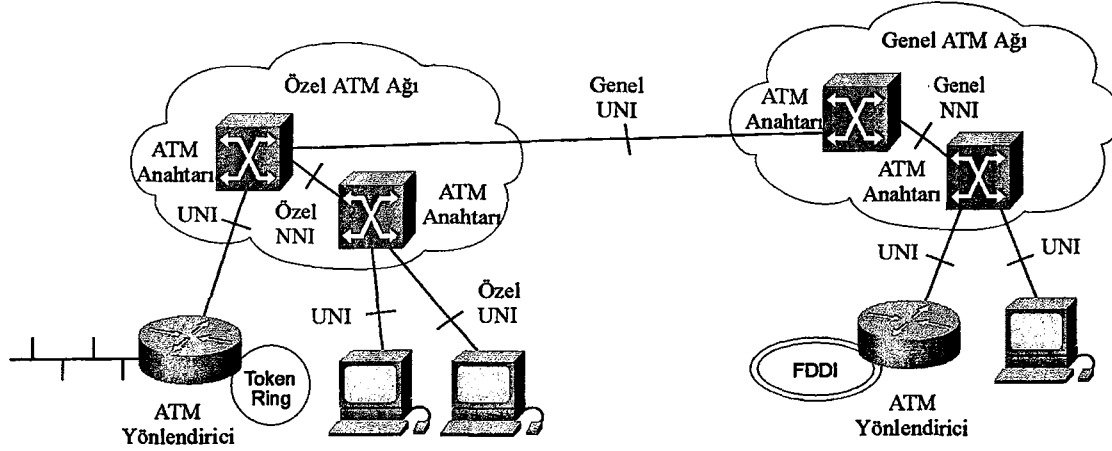
Ancak sunduğu tüm avantajlara rağmen ATM karmaşık bir teknolojidir. Her ne kadar ATM hücre yapısı ve hücre anahtarlama yüksek performanslı ATM anahtarlarının donanımsal olarak gerçekleştirilmesini kolaylaştırır da, ATM ağları, aynı zamanda, yazılım destekli oldukça kompleks protokol alt yapısına ihtiyaç duyar. Bu protokol alt yapısının hem bir ATM anahtarının bir ağa hem de ATM ağlarının mevcut diğer ağlarla olan bağlantılarına destek vermesi gerekmektedir.

ATM'nin bağlantı tabanlı olması, ATM belirli işaretleşme protokollerinin ve adresleme yapılarının yanı sıra bağlantı talebinin ağ boyunca yönlendirilmesinde kullanılacak protokollere gerek duyulduğu anlamına da gelir.

Bir ATM ağı, arayüzlerle veya noktadan-noktaya ATM bağlantılarıyla birbirlerine bağlı bir grup ATM anahtarından oluşur. ATM anahtarları iki türlü arayüz destekler: kullanıcı-ağ arayüzleri (UNI) ve ağ-düğüm arayüzleri (NNI). NNI, iki ATM anahtarını birbirine bağlayan bir arayüz olarak tanımlanır. UNI ise, ATM uç-sistemlerini (konak, yönlendiriciler vb.) bir ATM anahtarına bağlar; UNI ve NNI'da aralarında az fark olan hücre yapısı kullanılır. Daha açıkçası, NNI, iki ATM anahtarının üzerinde NNI protokolünü kullandığı fiziksel veya mantıksal bağıdır (Şekil 1.1).

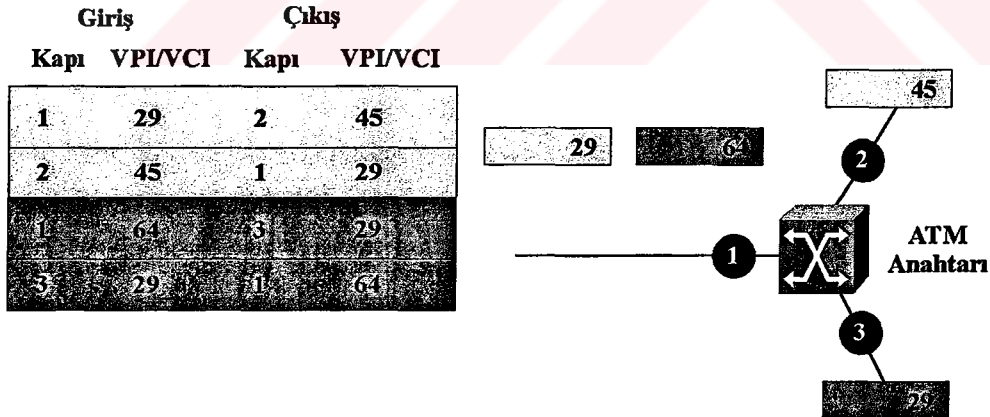
ATM ağları tamamen bağlantı tabanlıdır. Bunun anlamı, herhangi bir veri aktarımından önce sanal bir devrenin ATM ağına kurulması gerektiğidir.

ATM devreleri iki türdür: Sanal güzergahlar, sanal güzergah tanıttıcıları (VPI) ile belirtilirler; ve sanal kanallar, bir VPI ile sanal kanal tanıttıcısının (VCI) kombinasyonu ile belirtilirler. Sanal güzergah, sanal kanallar paketidir, bunların tümü ortak VPI temelinde ATM ağında saydam olarak anahtarlanırlar [1,2].



Şekil 1.1 ATM ağlarının birbirine bağlanması

Ancak tüm VPI ve VCI'ların bir bağda sadece yerel bir önemi vardır ve her anahtarda uygun şekilde yeniden eşlenir. Normal koşullarda düğümler tüm UNI bağlantılarını VPI=0 üzerinden yaparlar.

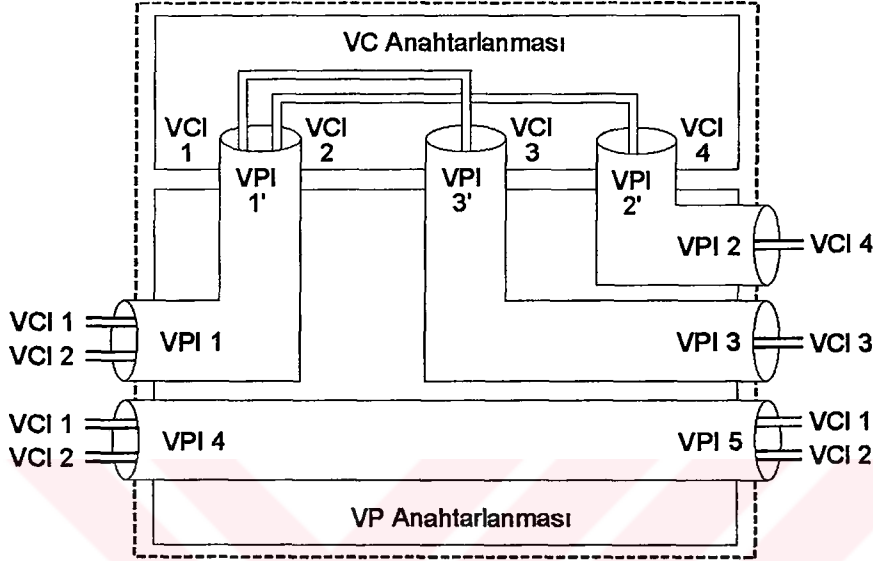


Şekil 1.2 ATM Anahtar Operasyonu

ATM anahtarının temel işlevi oldukça basittir; Bu işlev bilinen bir VCI veya VPI değeri üzerindeki bağ üzerinden hücre almak; bağlantının dışı doğru giden kapısını (veya kapılarını) ve bu bağ üzerindeki bağlantının yeni VPI/VCI değerini belirlemek

için yerel çeviri tablosunda bağlantı değerini aramak; ve daha sonra dışa doğru giden bu bağdaki hücreyi uygun bağlantı tanıtıcılarıyla tekrar göndermektir (Şekil 1.2).

Anahtar operasyonun bu kadar basit olmasının sebebi harici mekanizmaların herhangi bir verinin gönderilmesinden önce yerel çeviri tablolarını kurmasıdır.

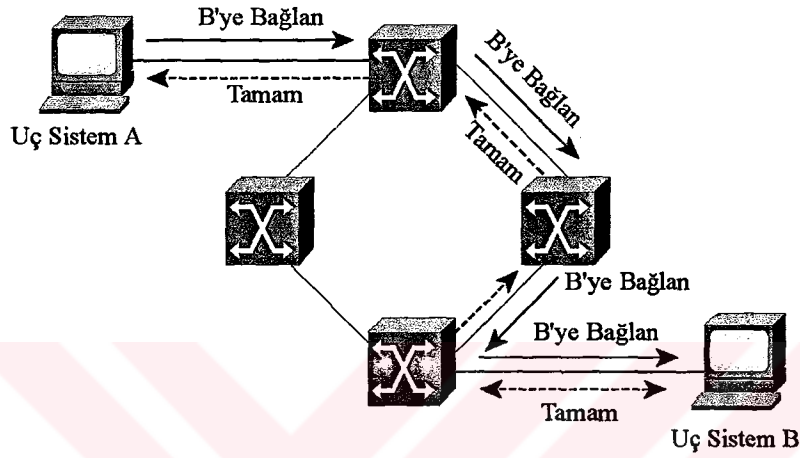


Şekil 1.3 Sanal Devre ve Sanal Güzergah Anahtarlanması

Bu tabloların oluşturulmasında kullanılan yöntem ATM bağlantılarının iki temel tipini belirler (Şekil 1.3):

- **Kalıcı Sanal Devreler (PVC):** PVC harici bir mekanizma ve tipik olarak ağ yöneticisi tarafından kurulan bir bağlantıdır. İçinde, kaynak ATM ve hedef ATM sistemleri arasındaki anahtarlar kümesi uygun VPI/VCI değerleri ile programlanır. İleride de ele alınacağı şekilde, ATM işaretlemesi bir PVC'nin kurulumunu kolaylaştırabilir, fakat PVC'ler, tanım gereği, her zaman bazı manuel konfigürasyonlara gereksinim duyarlar.
- **Anahtarlı Sanal Devreler (SVC):** SVC, işaretleme protokolü yoluyla otomatik olarak kurulan bir bağlantıdır. SVC'ler PVC'lerin kurulmasında gereken manuel etkileşime gerek duymazlar ve bu yüzden kullanımları çok daha fazla kabul görmüştür. ATM üzerinden çalışan tüm üst katman protokolleri SVC'ler kullanırlar.

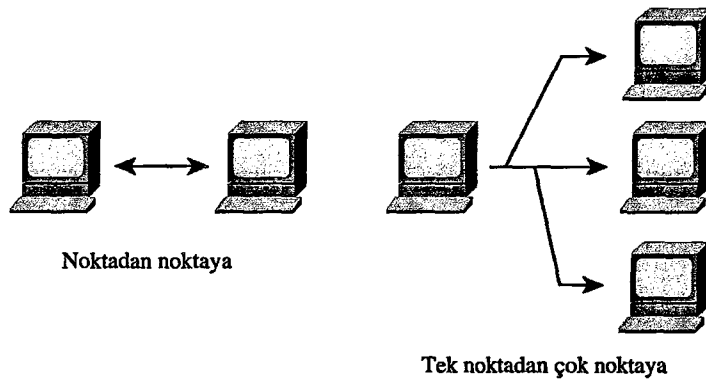
Bir ATM ağında ATM işaretleşmesi, bağlantı kurmak isteyen bir ATM uç-sistemi tarafından başlatılır; işaretleşme paketleri işaretleşme için rezerve edilmiş bir sanal kanala gönderilir (VPI=0, VCI=5). İşaretleşme ağ boyunca, anahtardan anahtara yönlendirilir, hedef uç sisteme ulaşana dek ilerledikçe bağlantı tanıtıcıları kurulur. Hedef uç sistem ya bağlantı talebini kabul edip onaylar ya da reddedip bağlantıyı temizler. Bağlantı, bağlantı talebinin güzergahından kurulduğu için veriler de aynı güzergahı takip ederler (Şekil 1.4).



Şekil 1.4 Bağlantı Kurulumu

ATM iki temel bağlantı tipini destekler (Şekil 1.5):

Noktadan noktaya bağlantılar, iki ATM uç sistemini bağlar. Bu tür bağlantılar tek-yönlü veya çift-yönlü olabilir.



Şekil 1.5 Bağlantı Tipleri

Tek noktadan-çok noktaya bağlantılar, bir kaynak uç sistemini (kök düğüm) çoklu hedef uç sistemlere bağlar (yapraklar). Hücre tekrarı, bağlantının iki veya daha fazla

dala bölündüğü ATM anahtarları tarafından ağ içinde yapılır. Bu tür bağlantılar tek yönlüdür, kökün yapraklarına geçişine izin verir ama yaprakların aynı bağlantıda köke veya birbirlerine geçişine olanak vermez.

ATM'nin bağlantı tabanlı yapısı, belirli işaretleşme protokollerinin ve adresleme yapılarının yanı sıra bağlantı talebinin ağ boyunca yönlendirilmesinde kullanılacak protokollere gerek duyulduğu anlamına da gelir. ATM Forum, ATM ağlarında kullanılabilecek NNI protokollerini tanımlamak amacıyla PNNI'yı geliştirmiştir.

PNNI protokolü iki bileşenden oluşur. Bunlardan birincisi; kaynak ve hedef UNI arasında, ATM bağlantı talebini ağ içinde aktarılmasında kullanılan PNNI işaretleşme protokolüdür. UNI işaretleşme talebi, kaynak anahtarda, NNI işaretleşmesine eşlenir. NNI işaretleşmesi ise, hedef anahtarda, UNI işaretleşmesine tekrar geri eşlenir.

ATM Forum tarafından geliştirilen, günümüz PNNI işaretleşme protokolü [3] UNI işaretleşmesinin geliştirilmiş bir halidir ve belirlenmiş geçiş listesi (DTL) gibi ek bilgi elemanlarını (IE) bünyesinde barındırır. PNNI işaretleşmesi, UNI işaretleşmesi içinde kullanılan sanal kanal VCI=5 üzerindeki NNI bağları üzerinden taşınır. VPI'nın değeri NNI bağının fiziksel veya sanal olmasına göre değişir.

PNNI protokolünün ikinci bileşeni ise sanal devre yönlendirme protokolüdür. Bu protokol, işaretleşme talebinin ATM ağı içerisinde yönlendirilmesi için kullanılır. Bu yönlendirmeye oluşturulan güzergah ATM bağlantısının kurulduğu ve verilerin akacağı güzergahtır. Bir ATM ağında işaretleşme talebinin yönlendirilmesi, ATM'in bağlantı tabanlı yapısına rağmen, diğer ağ katmanı protokollerindeki (IP gibi) bağlantısız paket yönlendirmesine benzerlik göstermektedir. Bunu doğuran sebep ise bağlantı kurulmasından önce, işaretleşme talebinin takip edeceği bir bağlantının bulunmamasıdır[4].

Bir VC yönlendirme protokolü, bağlantısız yönlendirme protokollerinin baz aldığı bir çok kavramı bünyesinde barındırabilir. Fakat PNNI protokolü varolan tüm yönlendirme protokollerinden daha komplekstir. Bu komplekslik protokolün sağlamak istediği iki ana amaçtan doğar; varolan diğer protokolden daha büyük ölçeklendirmeye izin vermek ve QoS ile yönlendirmeyi desteklemek.

Bu çalışmada esas olarak ATM anahtarları arasında ve ATM anahtar grupları arasında kullanılan PNNI protokolü, özellikle de bu protokolün yönlendirme mekanizması incelenmiştir. Bu amaçla ilk bölümde, ATM ağlarına ilişkin genel tanımlar verildikten sonra, ikinci bölümde PNNI'nin sahip olduğu hiyerarşik yapı, düğümler ve onların oluşturduğu peer gruplar arasında topoloji bilgilerinin dağıtılması konu edilmiştir.

Üçüncü bölümde ise PNNI İşaretleşme protokolü anlatılmıştır. ATM ağında bağlantı kurmak için gerekli mesaj akışları, bir bağlantı hatasının ortaya çıkabileceği durumlara karşı çok seçenekli yönlendirme, kaynaktan yönlendirme ve crankback'i destekleyecek mekanizmalar da bu bölüm içinde ele alınmıştır.

Dördüncü bölümde ise PNNI yönlendirme protokolünün sahip olduğu işlev ve mekanizmalar detaylanarak anlatılmıştır.

Beşinci bölümde, PNNI protokolünün başarımına etki eden faktörler incelenmiş, bu konularda halen sürdürülen çalışmaların şu anki durumları ve elde edilen sonuçları irdelenmiştir. Bunlara ek olarak, PNNI'nin 1996 yılında çıkan ilk sürümüne (PNNI v1.0) yapılan eklenti ve düzeltmelerin yanı sıra PNNI'nin çıkarılması düşünülen yeni sürümüne (PNNI v2.0) eklenecek kavram ve özellikler de ele alınmıştır.

Sonuç bölümünde, ATM Forum tarafından temelleri atılan PNNI protokolüne ait tasarım kararları, üretici firmaların inisiyatifine bırakılmış seçenekler incelenmiş ve mevcut diğer protokollerle olan karşılaştırmalar yapılmıştır.

## 2 PNİ YÖNLENDİRME PROTOKOLÜ

PNİ protokolü temel olarak iki protokolden oluşur: bunlardan ilki yönlendirme işlevlerinin yerine getirilmesi, ikincisi ise ağ boyunca gerekli işaretleşme taleplerinin idaresi içindir[5]. Bu bölümde PNİ yönlendirme protokolüne bir giriş yapılmaktadır.

PNİ yönlendirme protokolünün temel işlevleri şunlardır :

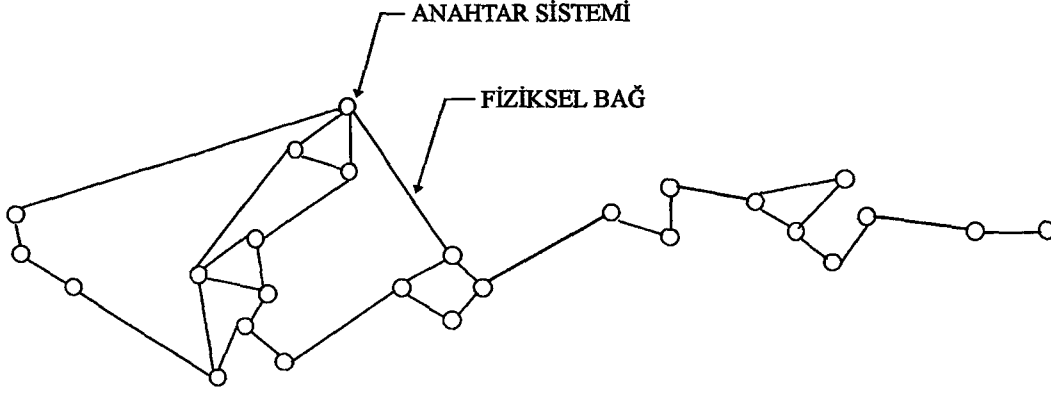
- Komşuların ve bağ durumunun keşfedilmesi
- Topoloji veri tabanlarının zamanuyumlaması
- PTSE'lerin taşkın yöntemiyle dağıtılması
- PGL'lerin seçilmesi
- Topoloji durum bilgisinin özetlenmesi
- Yönlendirme hiyerarşisinin kurulması

Bu işlevlerin anlaşılması amacıyla, sırasıyla protokolünün anlatımında gerekli tanımlar ve terminoloji verilecek ve temel işlevler tanıtılacaktır.

### 2.1 Fiziksel Ağ

PNİ yönlendirme protokolü en düşük seviyeli fiziksel düğümlerden oluşan ağlara uygulanır (Şekil 2.1).

Veri paketleri, bir en düşük seviyeli düğümden bir diğer en düşük seviyeli düğüme veya uç sistemlere geçerler. Uç sistemler, bağlantıların başladığı ve sonlandığı noktalardır. Güzergah belirlemek için uç sistemler ATM Uç Sistem Adreslerinin ilk 19 sekizlisiyle belirtilmiştir. Seçici sekizli (son sekizli) PNİ tarafından yön belirleme amaçlı kullanılmasa da uç sistemler tarafından kullanılır.



Şekil 2.1 Örnek ATM ağı

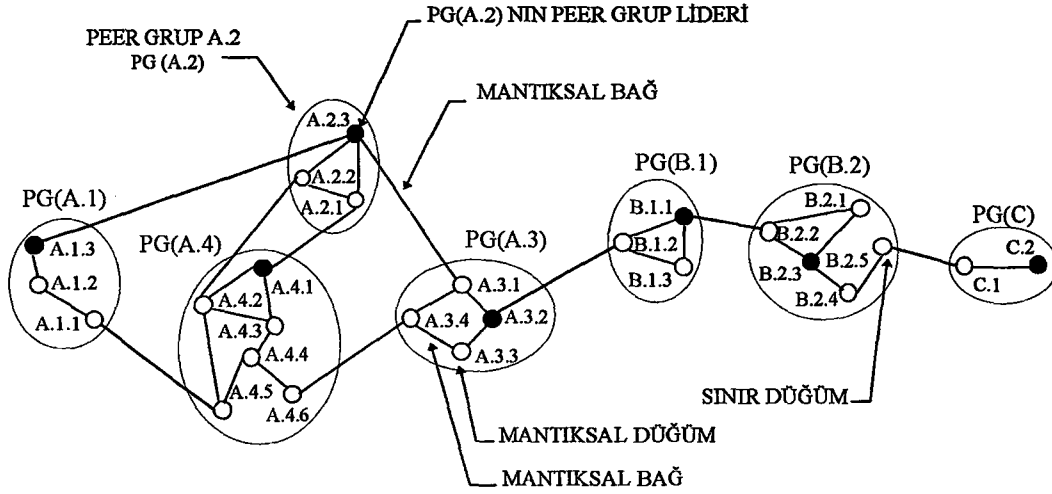
Şekil 2.1 deki her yay, iki anahtar sistemini birleştiren bir fiziksel bağı temsil eder. Kapı, bir anahtar sistemi içinde bağı en düşük seviyeli düğümle birleştiği noktadır. Fiziksel bağlarda trafik her iki yönde de akabilir. Fakat, fiziksel bağ karakteristikleri her iki yönde farklı olabilir, buna sebep ya kapasitelerinin farklı olması ya da var olan trafik yüklerinin farklı olmasıdır. Bu yüzden her fiziksel bağ, her biri bir yön için kullanılmak üzere iki küme farklı parametre ile gösterilir. Bu türde bir küme, bir iletim kapısı belirteci ile o kapıyı içeren en düşük seviyeli düğümün düğüm kimliğinden oluşur.

## 2.2 En Düşük Hiyerarşik Seviye

PNNI protokolü, sadece Şekil 2.1’de gösterilen düz ağ yapısını destekleseydi tüm en düşük seviye düğümler dahil (ağıdaki tüm fiziksel bağ için bilgiler ve ağıdaki tüm düğümler için ulaşılabilirlik bilgisi dahil) ağın tüm topolojisini barındırmak zorunda kalırdı. Bu yapı küçük spoke’lu ağlar için uygulanabilir olsa da büyük spoke’lu ağlar için çok büyük ek maliyetler doğururdu. PNNI yönlendirme hiyerarşisi, etkin bir yönlendirme sağlarken aynı zamanda bu türde ek yükleri de azaltmak üzere dizayn edilmiştir.

### 2.2.1 Peer Gruplar ve Mantıksal Düğümler

PNNI hiyerarşisi, düşük seviyeli düğümlerin peer gruplar olarak organize edildiği en alt düzeyde başlar. Hiyerarşi seviyesi bağlamında “mantıksal düğüm”, bir alt düzeye ait bir düğümdür. Daha anlaşılır kılmak için mantıksal düğümler genellikle “düğüm” olarak adlandırılırlar.



Şekil 2.2 PNNI hiyerarşisine kısmen sahip bir ATM ağı

Bir peer grup (PG), her biri grubun diğer üyeleri ile bilgi alış-verişi içinde bulunan mantıksal düğümlerin toplamından oluşur. Böylece tüm üyeler grubun özdeş görüntüsüne sahip olur.

Şekil 2.2 deki örnek ağ 7 peer grup şeklinde organize edilmiştir, bunlar : A.1, A.2, A.3, A.4, B.1, B.2 ve C. Düğümler ve peer grupları arasında ortaya çıkabilecek karışıklıkları önlemek amacıyla şekilde peer grupları PG( ) şeklinde gösterilmiştir. Örneğin, PG(A.3), A.3 peer grubunu temsil eder.

Bir peer grubun, kendine ait bir “peer grup kimliği” vardır. Peer grup kimlikleri kurulum aşamasında belirlenir. Komşu düğümler peer grup kimliklerini aralarında “Hello paketleri” vasıtasıyla değiştirirler. Düğümlerin aynı peer grup kimliğine sahip olmaları aynı peer grubuna ait oldukları anlamına gelir. Karşılıklı değiştirilen kimliklerin farklı olması ise, düğümlerin farklı peer gruplarına ait olduğunu gösterir.

Bir “sınır düğümü”, ait olduğu peer grubun dışına en az bir bağa sahip olan düğümdür. Bunun sonucu olarak farklı peer grup kimliklerine sahip komşu düğümler, ait oldukları peer gruplarının sınır düğümleridir.

Peer grup kimliği, ATM Uç Sistem Adresi üzerinde en fazla 13 sekizlik bir ön ek olarak belirlenir. Böylece peer grup kimliği, bahsedilen peer grubuna ait bir veya daha fazla düğümün adresine varsayılan ön ek olarak atanabilir.

### 2.2.2 Mantıksal Bağlar ve Bu Bağların Kurulumu

Mantıksal düğümler birbirlerine “mantıksal bağ”larla bağlanırlar. En düşük seviye düğümler arasındaki mantıksal bağlar, ya bir fiziksel bağ (Şekil 2.1’de gösterilenler gibi) ya da iki en düşük düğüm arasında bir VPC’dir. Aynı peer grubundaki en düşük seviye düğümler arasındaki bağlar toplanarak gösterilmezler. Örneğin, iki fiziksel bağ Şekil 2.1’deki aynı çift en düşük seviye düğümünü birleştirecekse, Şekil 2.2’de iki ayrı mantıksal bağ ile temsil edilmelidirler. Bir peer grup içindeki mantıksal bağlar “yatay bağlar”dır, diğer yandan iki peer grubu birleştiren bağlar “dış bağlar”dır.

Bir mantıksal bağ işler hale geldiğinde, bağlı düğümler “PNNI Yönlendirme Denetim Kanalı”(RCC) olarak kullanılan VCC yoluyla bilgi alışverişine başlarlar. Bu bağdaki düğümler tarafından periyodik olarak gönderilen Hello paketleri, ATM Uç Sistem Adresini, düğüm kimliğini ve bağlantı için kapı kimliğini belirtirler. Bu yolla Hello protokolü iki komşu düğümün birbirini tanımasını sağlar. Daha önce de bahsedildiği gibi PNNI Hello protokolü, peer grubu kimliklerin karşılıklı değiştirilmesini de destekler, bu sayede komşu düğümler aynı peer grubuna mı yoksa farklı peer gruplarına mı ait olduklarını belirlerler.

### 2.2.3 Veritabanının Oluşturulması

Her düğüm komşularıyla Hello paketleri değişir, ve böylece kendi yerel durum bilgisini oluşturur. Bu durum bilgisinin içinde düğümün en yakın komşusunun kimliği, peer grup üyeliği ve komşuları ile bağlantı durumu bulunur. Daha sonra her düğüm kendi durum bilgisini “PNNI Topoloji Durum Elemanları” (PTSE’ler) içinde toplar.

PTSE’ler, her peer grup içindeki mantıksal düğümler arasında dolaşan PNNI yönlendirme bilgilerinin en küçüğüdür. Bir düğümün topoloji veritabanı, alınan tüm PTSE’lerin içerdiği bilgilerin birleştirilmesiyle oluşur, bu sayede düğüm PNNI yönlendirme alanının o anki görüntüsünü oluşturur. Özellikle topoloji veritabanı, bir düğüme o yönlendirme alanından ulaşılabilir herhangi bir adrese gidecek güzergahın hesaplanması için gereken tüm bilgileri içerir.

### 2.2.3.1 D ğ m Bilgileri

Her d ğ m kendi kimliđini ve kaynaklarını, peer grup liderini se mek i in kullanılan bilgileri ve ayrıca PNNI hiyerar sisini olu turmak i in kullanılan bilgileri i eren bir PTSE olu turur. Bu bilgilere ise d ğ m bilgileri denir.

### 2.2.3.2 Topoloji Durum Bilgisi

PTSE'ler, diđerlerinin bilgilerin yanı sıra, "topoloji durum parametrelerini" de i erir ( rneđin mantıksal bađların karakteristiklerini belirten "bađ durum parametreleri", ve d ğ mlerin karakteristiklerini belirten "d ğ m durum parametreleri").

Topoloji durum parametreleri,  zellik ya da metriklerine g re sınıflandırılırlar.  zellik bilgileri, y nlendirme kararları verilirken ayrı olarak ele alınır.  rneđin, g venliđe ili kin bir "d ğ m  zelliđi" mevcut bir g zergahın reddine yol a abilir.  te yandan metrik bilgileri, etkileri g zergah boyunca birikimli olarak devam eden parametrelerdir.  rnek olarak gecikme metrikleri, verilen g zergah boyunca toplanarak elde edilir.

Bazı topoloji durum bilgileri,  zellikle bant geni liđi ile ilgili olanlar, zaman i inde daha  ok deđi kenlik g sterirler.  te yandan, y netimsel ađırlık gibi diđer topoloji durum bilgileri g receli olarak daha az deđi kendir. PNNI topoloji bilgilerinin dađıtımında kullanılan ta kın mekanizması dinamik ve statik topoloji durum bilgileri arasında fark g zetmez.

### 2.2.3.3 Topoloji Bilgilerinin Deđi imi

Kom u d ğ mler (mantıksal bađlantının her iki ucundaki d ğ mlerin birbirlerine Hello paketleri g ndermesinden sonra) aynı peer grubunda ait oldukları sonucuna varılırsa "topoloji veri tabanlarının" zamanuyumlamasını sađlamak  zere harekete ge erler. Veritabanı zamanuyumlaması, aynı topoloji veritabanına sahip olan iki d ğ mle kom u d ğ m arasındaki bilgi alı veri idir. Topoloji veritabanı, mantıksal d ğ m n bulunduđu peer grubu hakkında topoloji bilgileri, ve PNNI y nlendirme alanının arta kalan kısmını temsil eden  zet topoloji bilgileri i erir. Bu t rde y ksek seviyeli bilgilerin peer gruba hangi  ekilde aktarıldıđı 2.3 ve 2.4 b l mlerinde anlatılacaktır.

Topoloji veritabanı zamanuyumlaması süreci boyunca düğümler ilk olarak bağlantı bilgilerini içeren PTSE başlık bilgilerini karşılıklı olarak değiştirirler, böylece karşı tarafı kendi topoloji bilgilerini içeren PTSE’lerden haberdar ederler. Bir düğüm, sahip olduklarından daha yeni tarihli ya da henüz sahip olmadığı bir PTSE başlık bilgisi alırsa, bu PTSE için istekte bulunur ve topoloji veritabanını bu yeni PTSE ile günceller. Eğer yeni bir düğüm mevcut bir peer gruba katılırsa oluşacak veritabanı zamanuyumlaması, yeni düğüm açısından tek yönlü bir veritabanı kopyalaması şeklinde gelişir.

PTSE aktarımı yoluyla bir bağ tanıtımı ilgili komşu düğümler arasındaki zamanuyumlaması başarıyla bitirilmedikçe başlamaz. Bu yolla bağ durum parametreleri, bu bağı barındıran peer gruptaki tüm düğümlerin topoloji veritabanına dağıtılır. Bağların durum parametrelerinin peer gruptaki tüm düğümlere tanıtılmasında taşkın mekanizması kullanılır.

#### **2.2.3.4 Taşkın Yöntemi**

Taşkın mekanizmasının temeli, PTSE’lerin peer grup boyunca sıçramak suretiyle güvenilir olarak yayılmasıdır. Bir peer gruptaki tüm düğümlerin ortak bir topoloji veritabanına sahip olmalarını amaçlar. Taşkın, PNNI’nın kullandığı bir duyuru mekanizmasıdır.

Temel olarak taşkın prosedürü şu şekilde çalışır; PTSE’ler ileilmek üzere “PNNI topoloji durum paketleri (PTSP’ler)” şeklinde birleştirilirler. Bir PTSP alındığında içinde barındırdığı PTSE’ler incelenir. Her bir PTSE için, alınan PTSE’nin başlık bilgilerini de içeren, bir alındı paketi yollanır. Eğer alınan PTSE yeni ya da o düğümün elinde bulunan PTSE’sinden daha yakın tarihliyse topoloji veritabanına yerleştirilir ve PTSE’nin alındığı düğüm hariç, tüm komşu düğümlere taşkın yöntemiyle dağıtılır. Komşulara gönderilen PTSE’ler, alındısı alınana kadar periyodik olarak tekrar edilir.

Taşkın prosedürü sürekli olarak devam eder. Topoloji veritabanında bulunan PTSE’ler yaşlandırılırlar ve önceden belirlenmiş bir süre sonunda tazelenmezlerse topoloji veritabanından çıkarılırlar. Bir PTSE sadece, kendisini ilk olarak oluşturan düğüm tarafından tekrar oluşturabilir. PTSE’ler, hem periyodik olarak hem de olay tabanlı olarak yayınlanabilirler.

#### **2.2.4 Peer Grup Lideri Kavramı**

Bir peer grubun bir sonraki hiyerarşik seviyede, “mantıksal düğüm” adı verilen, tek bir düğümle temsili rolünü yüklenen düğüme “peer grup lideri” denir. Bu lider düğüm, temsil ettiği düğümler grubunun bir üyesidir. Her peer grubunda en fazla bir adet aktif grup lideri (PGL) bulunur.

PNNI hiyerarşisinin korunması için gerekli bilginin toplanması ve dağıtılmasındaki rolünün dışında PGL’nin peer grubunda başka bir belirgin rolü yoktur. Diğer tüm fonksiyonların gerçekleşmesinde, (örneğin bağlantı kurulması gibi) diğer sıradan düğümler gibi hareket eder.

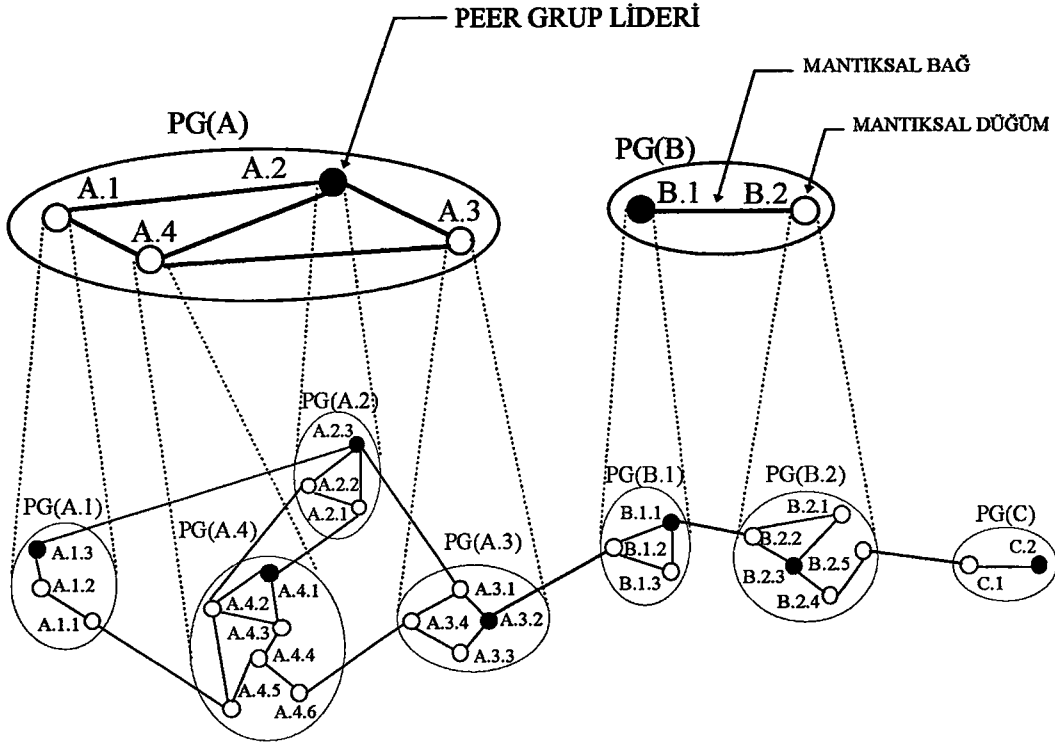
“Peer grup lideri seçimi” (PGLE) sonucunda, hangi düğümün peer grup lideri fonksiyonunu üstleneceği belirlenir. Seçim kriteri, düğümün “liderlik önceliği”dir. Bir peer grubunda en yüksek liderlik önceliğine sahip olan düğüm, o peer grubun lideri olur. Seçim işlemi sürekli devam eden bir prosedürdür. O anki grup liderinden daha büyük liderlik önceliğine sahip olan bir düğüm aktif hale geldiğinde, seçim işlemi yeni aktif olan düğümü peer grup liderliği konumuna getirir. Bir PGL düğüm ağdan çıkarıldığında ya da devre dışı kaldığında, liderlik önceliği bakımından hemen arkasından gelen düğüm yeni PGL olur. Eğer aynı liderlik önceliğine sahip birden fazla düğüm bulunursa, en büyük düğüm kimliğine sahip olan düğüm PGL seçilir. Bir düğüm PGL olarak seçildiğinde, grup içinde kararlı bir durumu garantilemesi için, düğümün liderlik önceliği artırılır.

### **2.3 Hiyerarşik Seviyelerdeki Gösterim**

Bu bölümde, oluşturulan peer grupların bir sonraki hiyerarşik seviyelerde nasıl temsil edildiği anlatılmaktadır.

#### **2.3.1 Mantıksal Düğüm**

“Mantıksal Düğüm”, bir peer grubun bir sonraki hiyerarşi seviyesinde soyut bir düğüm kullanılarak temsil edilmesidir. Örneğin, Şekil 2.3’de A.2 mantıksal grup düğümü A.2 peer grubunu bir üst seviye peer grubunda temsil etmektedir. Şekil 2.2, peer gruplarının bir sonraki peer grup hiyerarşisinde nasıl organize olduklarını göstermektedir.



Şekil 2.3 Kısmen yapılandırılmış PNNI hiyerarşisi

Bir mantıksal grup düğümünün fonksiyonları arasında kendi çocuğu olan peer grubuna ait bilgileri toplamak ve özetlemek, ve bu bilgileri dahil olduğu peer grubuna taşımak bulunmaktadır. Bir mantıksal grup düğümü ayrıca kendi peer grubundan aldığı bilgiyi taşkın yöntemiyle dağıtması için kendi çocuğu olan peer grubunun PGL'sine aktarır.

Bir mantıksal grup düğümü, düğümün temsil ettiği peer grubun kimliğini varsayılan olarak içeren bir düğüm kimliği ile belirler.

Bir peer grubun temsil ediliş şekli, peer grup liderinin kullandığı algoritma ve politikaya göre değişir. Şekil 2.3'deki mantıksal düğümler peer gruplar halinde organize olmuşlardır. Örneğin, A.1, A.2, A.3 ve A.4 mantıksal düğümleri, peer grup A'da organize olmuşlardır. A peer grubunun, lider seçme işlemi sonucu seçilmiş bir peer grup lideri vardır (A.2 mantıksal düğümü). A peer grubu, A.2, A.2, A.3 ve A.4 peer gruplarının "aile peer grubu" olarak adlandırılır. Diğer taraftan, A.1, A.2, A.3 ve A.4 peer grupları A peer grubunun "çocuk peer grupları" olarak çağrılırlar.

Bir peer grubun kimliğinin uzunluğu, o grubun PNNI hiyerarşisinin içindeki seviyesini belirler. Buna “seviye belirteci” denir. PNNI seviyeleri yoğun değildir, yani belli bir topolojide tüm seviyeler kullanılmaz.

### **2.3.2 Hiyerarşide Yukarı Yönde Bilgi Akışı**

Bir mantıksal düğüm, bir alt düzeyindeki tüm peer grubu temsil eder. İlgili peer grup lideri, alttaki peer grubun bir üyesi olarak, peer grubundaki tüm diğer düğümlerden topoloji durum bilgisi almıştır. Bu sayede peer grup lideri, mantıksal düğümü oluşturabilmek için gereken tüm bilgilere sahip olur. Bu durum, peer grup liderinin başlattığı mantıksal düğümüne bilgi aktarması olarak düşünülebilir. Yukarı yöndeki bu akış, topoloji bilgilerini birleşimini ve ulaşılabilirlik bilgilerini taşır. Ulaşılabilirlik bilgileri, alt seviye peer grubundan hangi adreslerin ulaşılabilir olduğunu gösterir (bkz. bölüm 2.6). Topoloji bilgilerinin birleşimi ise peer grup dahilinde yönlendirme kararları alabilmek için gerekli olan özet topoloji bilgilerini içerir.

Üst seviyelere sadece ihtiyaç duydukları özet bilgilerin yayılmasını sağlayan bir filtreleme fonksiyonu vardır. PTSE’ler hiçbir zaman hiyerarşide yukarı yönde ilerlemezler. Bunun yerine mantıksal düğümlerce özetlenmiş bilgilerden üretilmiş PTSE’ler peer grupta taşkınla dağıtılırlar.

### **2.3.3 Hiyerarşide Aşağı Yönde Bilgi Akışı**

Bilgilerin hiyerarşik seviyede aşağı yönde akması, alt seviye peer gruplarındaki düğümlerin PNNI yönlendirme alanı yoluyla ulaşılabilir tüm hedeflere yönlenebilmesi için gereklidir. Bu bilgiler ışığında güzergah belirlenir.

Her bir mantıksal düğüm, altındaki peer grubu bilgiyle besler. Aşağı yönde akan bu bilgiler, LGN’nin bulunduğu peer gruptaki diğer üyelerden aldığı PTSE’lerden oluşur. Peer grup liderine doğru akıtılan her bir PTSE, peer grup lideri tarafından peer grup içinde taşkın yöntemiyle dağıtılır. Bu sayede, peer gruptaki her düğüm daha üst düzeylerin görüntüsüne sahip olur. Özet olarak, PTSE’ler, bir peer grupta yatay olarak ve çocuk peer gruplarına doğru aşağı yönde akar.

### 2.3.4 Üst Bağlar

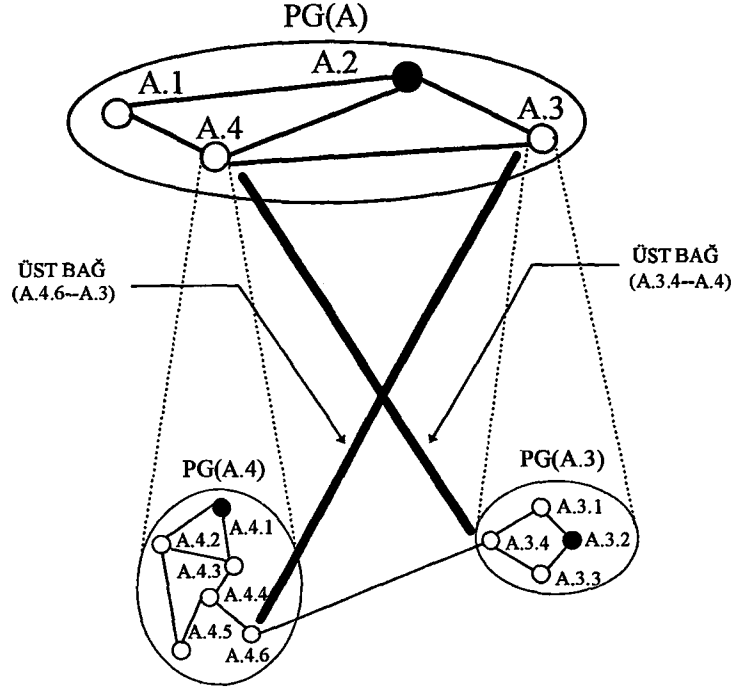
Hello protokolü sonrasında farklı peer gruplarına ait olduklarını anlayan komşu düğümler, sınır düğümler olurlar. Örneğin, A.3.4 ve A.4.6'daki düğümler Şekil 2.4'de sınır düğümlerdir. Farklı peer gruplara ait sınır düğümler arasındaki bağlara dış bağlar denir. Dış bağlar arasında veritabanı alışverişi yapılmaz; bu bağlar üzerinde sadece Hello protokolü kullanılır.

Sınır düğümleri Hello protokolünü dış bağlar boyunca yayarlar, bunun amacı kendi üst seviye peer grupları (düğüme ait hiyerarşi listesi) ve bu peer gruplarında kendilerini temsil eden mantıksal düğümler hakkında bilgi içermektir. Bu bilgiler sayesinde sınır düğümleri, her iki sınır düğüme ait en düşük peer grubunu belirler. Örneğin, Şekil 2.4'de A.3.4 ve A.4.6 düğümleri A peer grubunun her ikisinde ortak olduğunu belirlerler.

Bu mekanizma ile her düğüm kendi peer grubu içindeki topolojinin tamamını (düğümler ve bağlar dahil) ve üst seviye baba, büyükbaba vb. peer grubunun genel topolojisini (özet halinde) bilirler. Düğümün, hangi (sınır) düğümler ile hangi üst seviye düğümler arasında bağlantı olduğunu anlaması için, sınır düğümleri bağları bu üst seviye düğümlere bildirmeleri gerekir. Bu türde işlevi olan bağlara üst bağlar denir.

Düğüme ait hiyerarşi listesi sınır düğümlerine ortak üst seviye peer gruplarını belirlemede ve sınır düğümlerinin bağlanabileceği üst seviye düğümlerini tanımada gereken bilgileri sağlar. Örneğin, Şekil 2.4'de A.3.4 sınır düğümü komşusu olan A.4.6'nın A ortak baba grubunda A.4 mantıksal grup düğümü ile temsil edildiğini fark eder. Sonuçta A.3.4 kendisi ile üst düğüm A.4 arasında bir üst bağ tanır. Üst bağ (A.3.4- - A.4) ile gösterilir. Benzer şekilde A.4.6 düğümünün üst düğüm A.3'e (A.4.6- - A.3) üst bağı vardır.

Sınır düğümleri, kendi peer gruplarına gönderdikleri PTSE'lerde kendilerine ait üst bağları belirtirler. Bu sayede peer grubtaki tüm düğümler topoloji veri tabanlarını üst bağlarla güncellerler.



Şekil 2.4 Üst bağlar

### 2.3.5 Yönlendirme Denetim Kanalları

Komşu PNNI düğümlerinin, PNNI yönlendirme bilgilerini değiştirmek için kullandığı yönlendirme denetim kanalları vardır. Komşu düğümler, PNNI yönlendirme hiyerarşisinin en düşük seviyelerinde yönlendirme denetim kanalları için rezerve edilmiş bir VCC kullanırlar. Mantıksal düğümleri arasındaki yönlendirme denetim kanalı bir SVCC'dir. Bu SVCC'yi kurmak için gereken bilgi, mantıksal düğüm tarafından temsil edilen peer grubundaki üst bağlardan gelir.

Örneğin, A.3 ve A.4 arasındaki RCC'yi kurmak için şu aşamalar izlenir. Peer grup lideri A.3.2, A.3.4'den gönderilen, (A.3.4- -A.4) üst bağına anlatan PTSE'yi alır. A.3.2, bu PTSE'den iki anahtar bilgi çıkarır; ortak peer grubunun kimliği, ve üst düğüm A.4'ün ATM Uç Sistem Adresi. A.3.2 düğümü bu bilgiyi A.3 mantıksal düğümüne iletir. A.3 bu bilgidan A.4 ile aynı peer grubunda olduğu sonucunu çıkarır. Çıkardığı başka bir sonuç da A.4'ün bir komşu olduğudur ve bu yüzden A.4'e verecek bir RCC'ye ihtiyaç duyar.

A.4'e ulaşılmakta kullanılan ATM Uç Sistem Adresi A.3'de olduğundan, SVCC'yi kurmak için yeterli bilgisi vardır. Benzer şekilde, A.4 mantıksal düğümü A.3'e SVCC kurmak için yeterli bilgiye sahiptir.

### 2.3.6 Yatay Bağlar

Yatay bağlar, aynı peer grup içindeki düğümler arasındaki mantıksal bağlardır. Şekil 2.4’de sadece (A.3.4- -A.3.3) yatay bağ değildir, (A.3- -A.4) de bir yatay bağdır.

Yatay bağlar, komşu düğümler ile Hello paketlerinin başarıyla değişimi ve veritabanının zamanuyumlaması sağlanmadan tanınmaz. Bu şartlar sağlandıktan sonra yeni bağı tanımlayan PTSE’ler artık peer grup içinde taşkın yöntemiyle dağıtılırlar.

Bir mantıksal düğüm çifti arasındaki tanımlanan yatay bağ, bu iki düğüm arasındaki yönlendirme amaçlı bağlanabilirliği gösterir. Mantıksal düğümler, yatay bağlara ve düğüme bağlı diğer bağlara kapı kimlikleri atamakla sorumludurlar.

### 2.3.7 Topoloji Bilgilerinin Birleştirilmesi

Topoloji Bilgilerinin Birleştirilmesi, büyük spoke’lu bir ağda ölçeklenebilirliğin sağlanabilmesi için gereken bağ ve düğüme ait bilgilerin indirgenmesini amaçlar. Tek amacı karmaşıklığı azaltmak değildir, aynı zamanda güvenlik sebebiyle peer grupları arasındaki iç topoloji bilgilerini gizlemeyi de amaçlar.

#### 2.3.7.1 Bağ Bilgilerinin Birleştirilmesi

Bağ Bilgilerinin Birleştirilmesi, aynı iki peer grup arasındaki bağ kümelerinin tek bir mantıksal bağ ile gösterilmesidir. Örneğin, Şekil 2.3’de düğüm A.2’yi A.4’e birleştiren bağ, iki bağı (A.2.1- -A.4.1) ve (A.2.2- -A.4.2), birleşimini temsil eder.

Mantıksal düğümler bağ bilgilerinin birleştirilmesinden sorumludur. Bir mantıksal düğüm, kendi çocuk peer grubundan belli bir üst düğüme kadar tüm üst bağ ilanlarını inceler. Toplanma sembolü aynı olan tüm üst bağlar tek bir bağa toplanırlar.

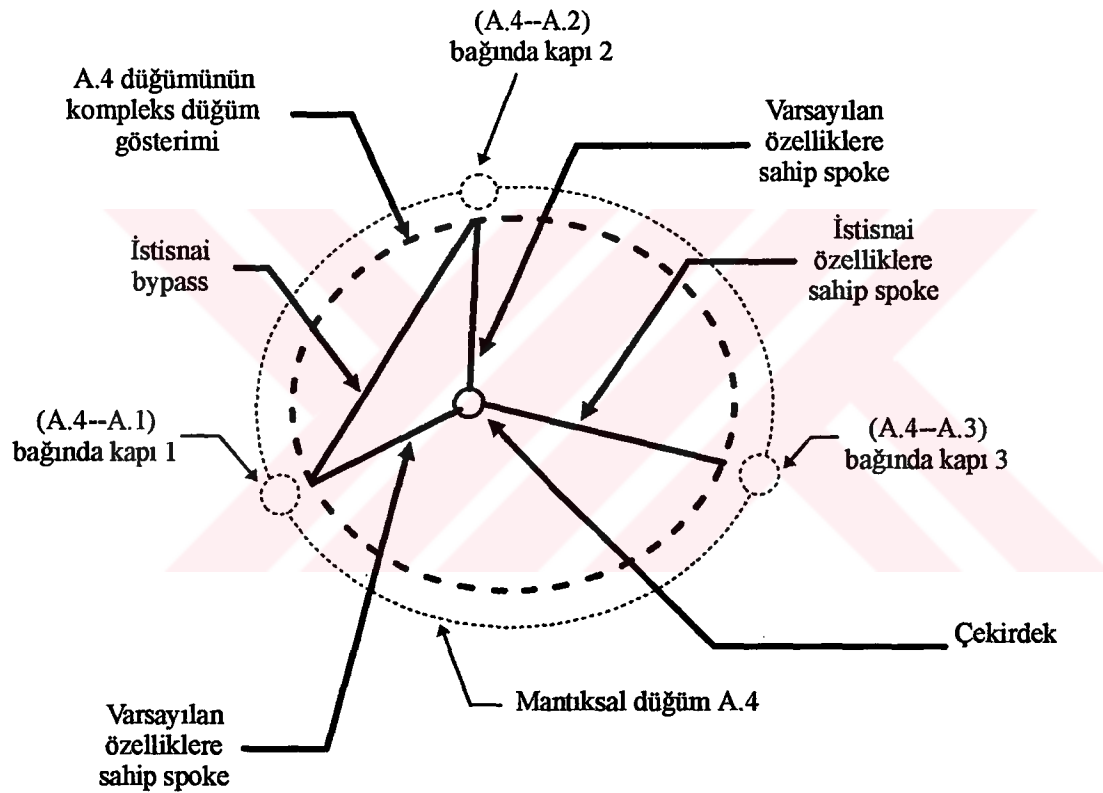
#### 2.3.7.2 Düğüm Bilgilerinin Birleştirilmesi

Düğüm bilgilerinin birleştirilmesi, bir çocuk peer grubunun baba peer grubunda bir mantıksal düğüm ile temsil edilmesidir.

Bu birleşimin baba peer grubundaki sonuçlarını temsil etmek için kompleks düğüm temsili kullanılır. En basit kompleks düğüm temsili, sabit yarı spoke’lu simetrik yıldız topolojisidir. Yıldızın merkezi, mantıksal düğümün iç referans noktasıdır ve

çekirdek adı ile anılır. Çekirdek ile mantıksal düğümün bir kapısı arasındaki bağlanabilirliğe de spoke denir. Simetrik yıldız topolojisi, herhangi bir mantıksal düğüm girişi veya çıkışı ile çekirdek arasında varsayılan değeri veren her bir düğüme ait durum parametresi için tek değerden oluşan bir düğüm temsili olarak kullanılır.

Genellikle peer gruplar simetrik değildir. Örneğin, “uzak elemanlar” içerebilirler. Bu asimetri bir “istisna” kümesi ile gösterilebilir. İstisnalar, çekirdeğe bağlanabilirliği varsayılandan oldukça farklı olan belirli kapıları temsil etmek için kullanılabilir. Buna ek olarak, bir istisna iki kapı arasında bağlanabilirliği temsil etmek içinde kullanılabilir.



Şekil 2.5 Şekil 2.3'deki LGN A.4'ün kompleks düğüm gösterimi

Kompleks düğüm gösterimine bir örnek olarak Şekil 2.3'teki A.4 peer grubunun kompleks düğüm gösterimi olan Şekil 2.5 verilebilir. Şekil 2.5'de çekirdek, A.4 mantıksal düğümünün içini temsil eder. Çekirdekten çıkan her spoke, A.3 mantıksal düğümünün bir kapısı ile ilişkilidir. Şekil 2.5'de her biri bir kapı için olmak üzere üç spoke bulunmaktadır. Bunlar :

- Kapı 1, (A.4- -A.1) bağı üzerindeki kapı

- Kapı 2, (A.4- -A.2) bağı üzerindeki kapı
- Kapı 3, (A.4- -A.3) bağı üzerindeki kapı

Şekil 2.5’de kapı 1 ve 2’nin spoke’ları varsayılan özellikleri kullanmakla beraber kapı 3’e ait spoke istisna teşkil etmektedir. Bu istisnanın sebebi, A.4.5 boyunca A.4.1 düğümleri yakın olarak kümelenmişken A.4.6’nın uzak bir eleman olması olabilir.

Düğüm geçişleri, çekirdeğinden geçen bir istisnaya da şekillenebilir. Örneğin, Şekil 2.5’de, kapı 1 ile kapı 2’yi birleştiren İstisna Bypass, A.1 ve A.2 düğümleri arasında giderken geçen A.4’e tekabül eder. Bu istisnaya yol açan sebep A.4.2 ve A.4.5 düğümleri arasındaki çok yüksek hızlı bir bypass bağ olabilir.

A.4 düğümü, kendi kompleks düğüm gösterimini PTSE’lerin A.1, A.2 ve A.3’e göndermek yoluyla dağıtır. PTSE’ler sonra kendi peer grup liderlerine geçerler, bunun sonucu olarak A.1, A.2, A.3 ve A.4 peer gruplarındaki tüm düğümlerin topoloji veri tabanları A.4’ün kompleks düğüm sunumunu içerir.

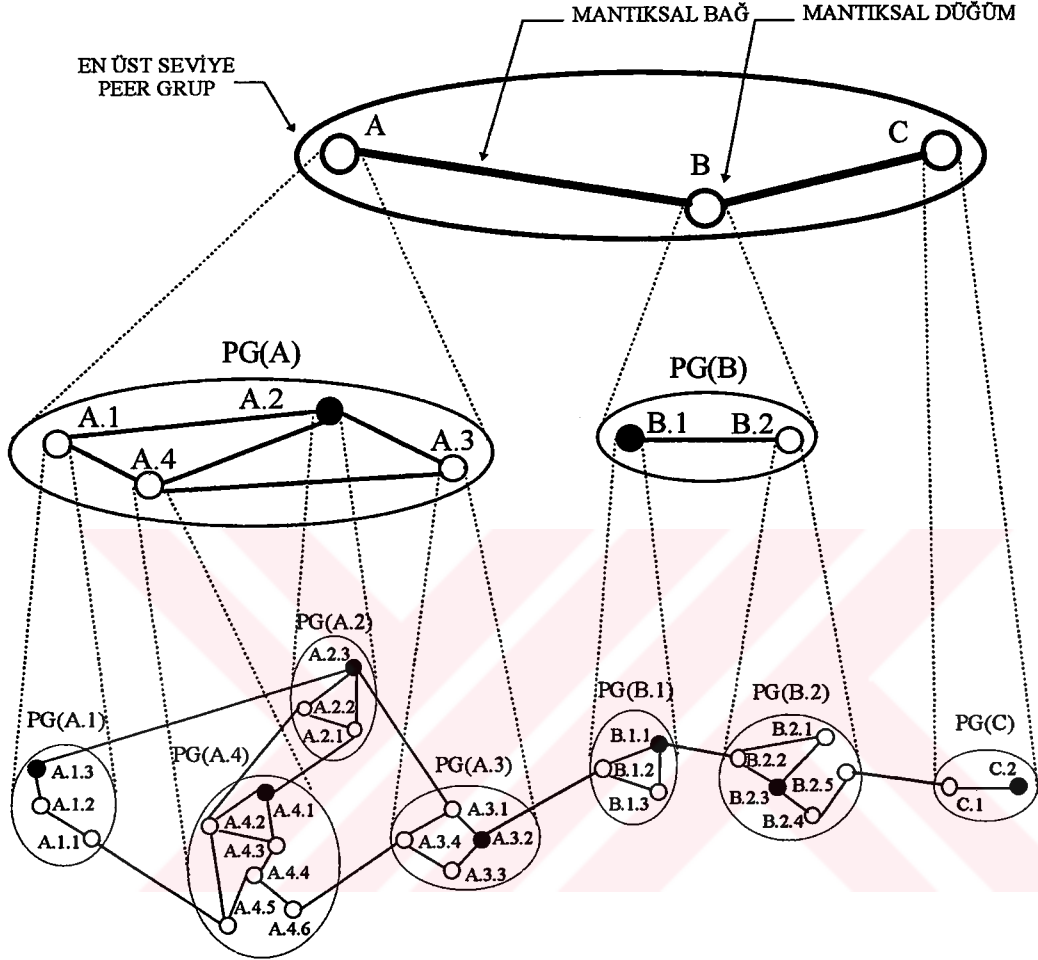
Bir mantıksal düğüm üzerinde yönlendirme, spoke’ların ve bypass’ların birbirine hangi şekilde bağlanacağını belirlenmesi şeklinde gelişir. Bağlantı, çağrının gereksinimlerini karşılayacak şekilde seçilmelidir. Örneğin (Şekil 2.3 ve Şekil 2.5) (A.1- -A.2) mantıksal bağının tıkanık olduğunu ve bu yüzden A.3.4 düğümünün ya (A.3- -A.4), (A.4- -A.1) ya da (A.3- -A.2), (A.2- -A.4), (A.4- -A.1) mantıksal bağları yoluyla A.1 peer grubunda bir düğüme yönlendiğini varsayalım. Dahası yönlendirme kriteri açısından iki güzergahın eşdeğer olduğunu farz edelim. Böyle bir durumda güzergah seçimi, olası en iyi üç kompleks düğüm geçişlerinden birini seçme anlamına gelir :

1. Varsayılan davranışlı iki spoke’un birbirine bağlanması
2. İstisna spoke ile varsayılan davranışlı bir spoke’un birbirine bağlanması
3. İstisna bypass

Eğer en iyi seçim İstisna bypass ise, bu durumda (A.3- -A.2), (A.2- -A.4), (A.4- -A.1) tercih edilen güzergah olacaktır. Bunu sonucu olarakta, A.3.4 düğümü A.1 peer grubunun üyesi bir düğüme yönelirken (A.4.2- -A.4.5) bağı kullanacaktır.

## 2.4 Yönlendirme Hiyerarşisinin Tamamlanması

Bu bölümde peer grupların kendini tekrarlayan yapısı üzerinde durularak PNNI yönlendirme hiyerarşisi tanımı yapılacaktır.



Şekil 2.6 Hiyerarşik olarak tamamen yapılandırılmış bir ağ

Önceki bölümlerde verilen PNNI yönlendirme hiyerarşisi örnekleri tamamlanmış bir hiyerarşik yapıya sahip değillerdi. Bunun sebebi ise, üst seviyeli peer gruplarının (Şekil 2.3) birbirleri arasında bağlanabilirlik göstermemeleridir. Ağın hiyerarşik olarak tamamlanması, daha da yüksek seviyeli peer gruplarının yaratılmasıyla elde edilmiştir, bu işlem ağın tamamının tek bir en yüksek seviye peer grubu tarafından kapsanmasına kadar devam eder. Şekil 2.6'daki örnekte bu, A, B ve C mantıksal grup düğümlerini içeren bir peer grubun daha yapılandırılması ile elde edilmiştir. A düğümü A peer grubunu temsil eder, A peer grubu da A.1, A.2, A.3, A.4 vb. peer gruplarını temsil eder. Olası bir diğer yapılanma, B ve C peer grupları BC peer grubu

şeklinde birleşseydi ortaya çıkabilirdi. BC peer grubu daha sonra en yüksek seviye peer grubunu oluşturmak üzere A peer grubu ile birleşecekti.

Oluşan hiyerarşik yapı oldukça esnektir. Birbirini takip eden peer grup sayısında üst sınır 104'e eşittir ki bu rakam gerekenden fazladır, çünkü uluslararası ağlar bile 10'dan az sayıda hiyerarşik seviyeyle bile yapılandırılabilirler.

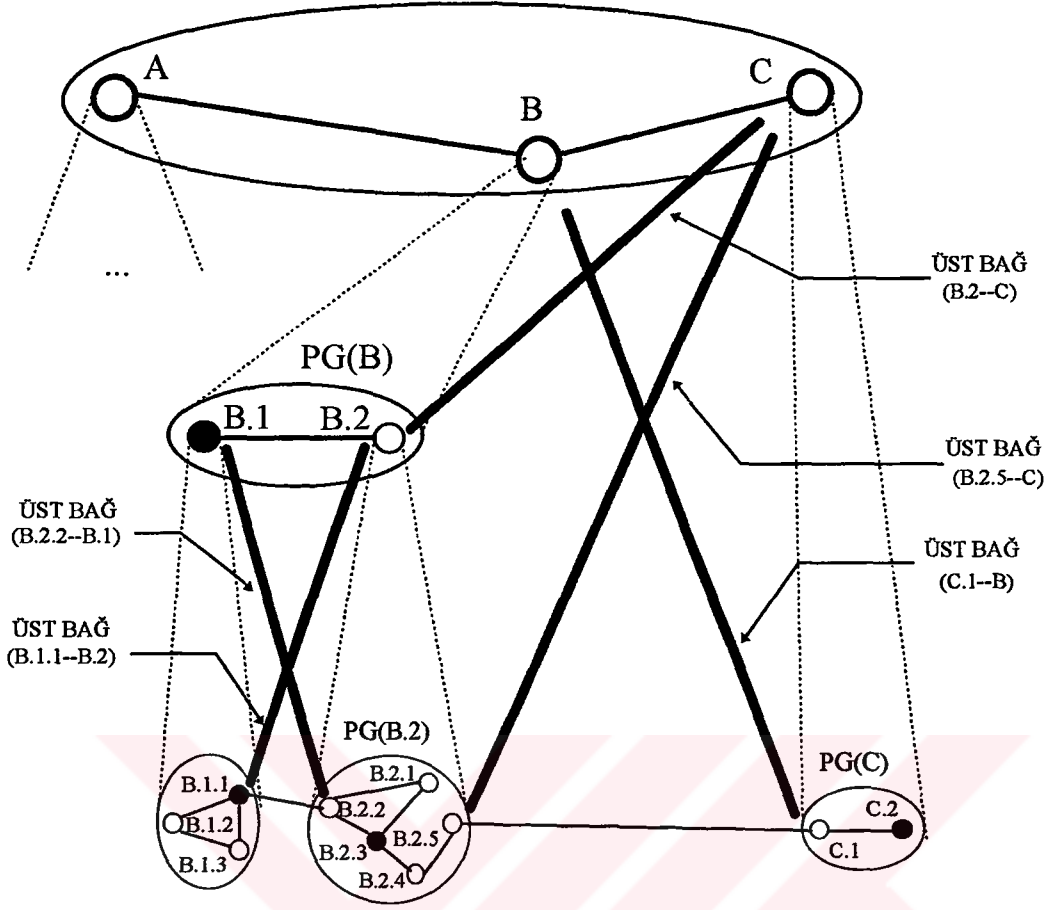
PNNI yönlendirme hiyerarşisi içinde asimetrliler oluşabilir; düşük seviyeli peer grubun baba peer grubu, aynı anda bir başka düşük seviyeli peer grubunun büyük babası veya büyük-büyük babası olabilir. Örneğin, Şekil 2.6'daki düşük seviyeli peer grup C, doğrudan doğruya en yüksek seviye peer grubunda mantıksal grup düğümü C ile temsil edilmiştir, ne var ki B.1 ve B.2 düşük seviyeli peer grupları, B mantıksal düğümü ile en yüksek seviyede temsil edilmeden önce B baba grubunda gruplaşmışlardır.

Bu senaryodaki üst bağlar Şekil 2.7'de gösterilmiştir. Dört üst bağın yaratılması (B.2.5- -C), (C.1.B), (B.1.1- -B.2) ve (B.2.2- -B.1), bölüm 2.3.4'te ele alınmıştır. Ama, (B.2- -C) üst bağı, (B.2.5- -C) üst bağından türemesi sebebiyle farklıdır. Buna istemli üst bağ denir.

Peer grup lideri B.2.3, (B.2.5- -C) üst bağı anlatan PTSE'ler aldığı anda (B.2.5 tarafından gönderilen) ortak peer grubu kimliğini (bu durumda en yüksek seviye peer grubu) ve üst düğüm C'nin ATM Uç Sistem Adresini LGN B.2'ye aktarır. Bu bilgiyi kullanarak B.2, C düğümünün B peer grubunun bir üyesi olmadığını anlar ve B.2'nin C'ye ulaşılabilirliği anlatan yeni bir üst bağ (B.2- -C) türetir. B.2, B.2.5'i temsil ettiğinden, (B.2- -C) (B.2.5- -C)'nin üst seviyedeki temsilidir.

Eğer B.2.1'i C.1'e bağlayan başka bir bağ bulunsaydı, bu durumda (B.2- -C) üst bağı iki düşük seviye üst bağın (bu bağlar (B.2.5- -C) ve (B.2.1- -C)) birleşimini temsil edecekti.

B ve C LGN'leri arasında bir SVCC-tabanlı RCC şu şekilde kurulur: (B.2- -C) üst bağı yaratıldıktan sonra B.2, yeni üst bağı belirten bir PTSE paketini B peer grubunda taşkın yöntemiyle dağıtır. Üst bağ bilgisini aldıktan sonra peer grubu lideri B.1, ortak peer grubu kimliğini (bu durumda en yüksek seviyeli peer grubu) ve C üst düğümünün ATM Uç Sistem Adresini LGN B'ye aktarır.



Şekil 2.7 Üst bağlar

Şekil 2.7’de B düğümü, üst bağı olduğunu belirten sınır düğümlerinden birini kullanarak, C düğümüne bir güzergah seçer. Şekil 2.7’de B.2, üst düğüm C’ye bir üst bağı sahip tek sınır düğümüdür ve bu yüzden LGN B ve LGN C arsında kurulacak SVCC B.2 sınır düğümünü kullanır.

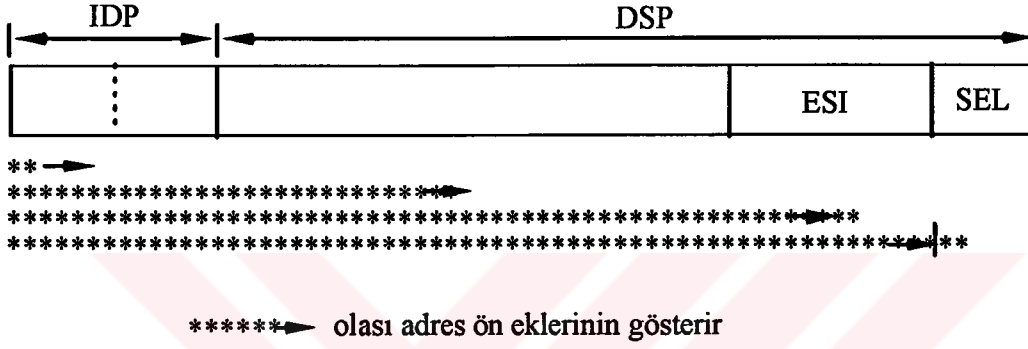
Sonuç olarak, PNNI yönlendirme hiyerarşisinin kuruluşuna, peer gruplarının yinelenerek üretimi gözüyle bakılabilir. Bu hiyerarşinin kurulmasına en düşük seviye düğümlerle başlanır ve tüm PNNI yönlendirme alanını kaplayan tek bir en üst seviye peer grubunun oluşturulmasıyla son bulur. Hiyerarşinin nasıl yapılandırılacağı, hangi Peer grup kimliklerinin hangi mantıksal düğümle ilişkilendirileceği yapılan konfigürasyonla belirlenir. Bu perspektiften bakıldığında PNNI yönlendirme hiyerarşisinin anahtar elemanı peer grubun yapısıdır.

Genellikle, bir peer grubun davranışı, bulunduğu seviyeden bağımsızdır. Bunun sonucu olarak tüm peer grupları için aynı tanımlar geçerlidir. Ama, en yüksek

seviyeli peer grup, bir baba peer grubunda, kendini temsil eden bir peer grup liderine gerek duymadığından diğer peer gruplarından farklıdır.

## 2.5 Adresleme Yapısı

PNNI yönlendirme hiyerarşisinin içeriğinin adreslenmesi ve tanımlanması, ATM Uç Sistem Adreslerine uygulanan ön eklerinin kullanımına bağlıdır. Şekil 2.8’de gösterildiği gibi, ATM Uç Sistem Adresleri, NSAP adresleme yapısıyla oluşturulurlar.



Şekil 2.8 NSAP Adresleme Yapısı ve Adres Ön Ekleri

ATM Uç Sistem Adresleri 20 sekizli uzunluğundadır. PNNI yönlendirme işlemi ATM adresinin sadece ilk 19 sekizlisinde uygulanır. Seçici sekizlinin (20inci) uç sistemde sadece yerel önemi vardır ve bu sebeple de PNNI yönlendirmesi tarafından kullanılmazlar. NSAP’ler genellikle adres yapısını belirleyen bir otorite tarafından atanırlar. PNNI, istisnalar dışında, ek bir adres yapısı kullanmaz veya yorumlamaz. İlk sekizli, Yetkilendirme ve Format Tanıtıcısı (AFI), tek bir ATM uç sistemini veya bir veya birden fazla ATM uç sistemini temsil eden grup adreslerini ayırt etmek için PNNI tarafından kullanılır. NSAP adresinin alt yapısı olan (IDP, DSP, ESI ve SEL) bütünlük sağlamak için dahil edilmiştir ama yönlendirme operasyonları için önemli değildir.

Bir ATM Uç Sistem Adresinin ön eklerinden biri, adresin ilk “p” bitidir. P’nin değeri 0 ila 152 bit arasında değişebilir. Sıfır uzunluğu olan bir ön ek, “tüm” ATM Uç Sistem Adreslerine tek bir ilanda ulaşır. Sıfırdan uzun ve 152’den kısa uzunluğu olan ön ekler adres alanının bazı kısımlarına ulaşmak için kullanılır. Kısa ön ekler daha büyük, uzun ön ekler ise daha küçük alanları kapsar.

Muhtemel bir adres grubu ön ekleri Şekil 2.8’de yıldızlı oklarla (\*\*\*\*\*>) gösterilmiştir. Oklar, ön ekler gibi adresin en sol ucundaki bitlere uygulanır. Ön ek uzunluğu bu okların ne kadar sağa uzayacağını belirler.

PNNI, topolojik olarak hiyerarşik yapıli ağlarda çalışır. Hiyerarşinin yapısı, yönlendirme alanında kullanılan, peer grup kimliği ile tanımlanır. Adres atanmasının, daha iyi ölçeklenebilirlik sağlanması için, genellikle topolojik hiyerarşiye uyan bir yapısı vardır. Bu sayede, adresin, belirtilen ön ekle başlayan tüm adreslere ulaşılabilirliğini temsil eden adres özetlenmesi mümkün olur.

### **2.5.1 Düğüm Adresleri**

PNNI yönlendirmesinde aktif olarak görev alan düğümler ATM Uç Sistem Adresleri kullanılarak adreslenirler. Tek bir anahtar sistemi birçok düğüm içerebilir. Bu düğümlerden her biri birer tekil ATM Uç Sistem Adresine ihtiyaç duyar. Bu adres, PNNI’da kullanılan SVCC’lerin kurulumunda kullanılır.

### **2.5.2 Uç Sistem Adresleri**

ATM Uç Sistem Adresleri 20 sekizliden oluşur. Yukarıda da anlatıldığı üzere PNNI yönlendirmesi ilk 19 sekizliyi kullanır. Seçici sekizli sadece aynı arayüzden ulaşılabilir hedefleri ayırt etmekte kullanılır.

Genellikle PNNI, uç sistemlere ulaşılabilirliği, özet oluşturmak için, ATM Uç Sistem Adreslerindeki ön ekleri kullanarak ilan eder. Eğer bir düğüme bağlı bir uç sistem, düğümün konfigüre edilmiş özetlerinden birine uymazsa bu durumda düğümün o uç sistem için açık bir ilan oluşturması gerekebilir. Bu ilan 152-bit (19 sekizlinin tamamını) taşıyacaktır. Uç sistemin ulaşılabilirlik özetinin ön ekleri, 152 bitten kısa olacaktır. Adresleme hiyerarşisinin topoloji hiyerarşisini takip ettiği durumlarda tek bir ön ek kullanarak birçok uç sisteme ulaşılabilirlik ilan etmek mümkündür. Kısa ön ekler daha fazla adres özetlerler.

## **2.6 Adres Özetlenmesi ve Ulaşılabilirlik Bilgileri**

Adres özetlenmesi, PNNI ağında dağıtılması gereken adres bilgilerinin miktarını azaltır ve büyük ağların desteklenmesi için gerekli ölçeklenebilirliğe katkıda bulunur. Adres özetlemesi, basit bir “ulaşılabilir adres ön eki” kullanılarak bir grup uç sistem

veya düğümün, belirtilen ön ekle başlayan adreslerinin temsilini amaçlar. Ulaşılabilir adres ön ekleri, ya özet adresler ya da yabancı adresler olurlar.

Bir düğüme ait “özet adres”, ya o düğümde açıkça yapılandırılmış ya da varsayılan değeri alan bir ön ektir. Düğümle ilgili “yabancı adres”, düğümün özet adreslerinden herhangi biriyle uyuşmayan adrestir. Bunun tersi olan “yerli adres” de düğümün özet adreslerinden herhangi biriyle uyuşan adrestir.

Şekil 2.6’dan türetilen Şekil 2.9’da, A.2.1, A.2.2 ve A.2.3 düğümlerine eklenen düğümler uç sistemleri temsil ederler. Her bir uç sistemle ilgili alfa-nümerik, o uç sistemin ATM adresini temsil ederler. Şekil 2.9, <A.2.3.2> bir ATM adresini, ve P<A> aynı ATM adresinin kısa ön ekini göstermektedir.

A.2 peer grubundaki her düğüm için yapılandırılmış özet adreslerin bir örnek listesi aşağıda gösterilmiştir :

Tablo 2.1 Yapılandırılmış Özet Adresler

| A.2.1’de özet adresler     | A.2.2’de özet adresler  | A.2.3’de özet adresler    |
|----------------------------|-------------------------|---------------------------|
| P<A.2.1> (yapılandırılmış) | P<Y.1>(yapılandırılmış) | P<A.2.3>(yapılandırılmış) |
| P<Y.2> (yapılandırılmış)   | P<Z.2>(yapılandırılmış) |                           |

Tablo 2.1’deki özet adreslerden farklı özet adreslerde seçilebilirdi. Örneğin, Şekil 2.9’da, A.2.2 düğümünde P<Y.1> in yerine P<Y.1.1> veya A.2.1’de P<W> özet adresleri kullanılabilirdi. Ama P<A.2.1> veya P<A.2.3> yerine P<A.2>, A.2.1 ve A.2.3 düğümlerinde varsayılan özet adres olarak seçilemezdi. Bunun sebebi ise güzergah belirleyen uzak bir düğümün, A.2.3’e bağlı uç sistemlerle A.2.1’e bağlı uç sistemleri ayırt edemeyecek olmasıdır.

Tablo 2.1’de, A.2.1’de belirtilen özet adres listesine göre P<W.1.1.1> yabancı bir adrestir, çünkü A.2.1’deki özet adreslerden hiçbirisiyle uyuşmaz.

LGN A.2, kendi özet adres listesine gereksinim duyar. Bu özet adres listesinin oluşturulmasında da alternatifler vardır. PG(A.2), A.2 peer grubunun kimliği olduğundan P<A.2>’yi varsayılan özet adres olarak katmak mantıklıdır. Buna ek

**A.2 peer grubunda, peer grup lideri olma potansiyeline sahip her düğüm aynı özet adres listesine sahip olmalıdır.**

|                                |
|--------------------------------|
| LGN A.2'nin Özet Adres Listesi |
| P<A.2> (varsayılan)            |
| P<Y> (yapılandırılmış)         |

Tablo2.3’de, her düğüm tarafından A.2 peer grubuna tanıtılan ulaşılabilir adres ön ekleri gösterilmektedir :

A.2.1 düğümü özet adreslerini ve yabancı adreslerini aktarırken, A.2.2 ve A.2.3 düğümleri sadece özet adreslerini sürerler çünkü yabancı adrese sahip uç sistemleri yoktur.

Tablo 2.3 Tanıtılan Ulaşılabilir Adres Ön Ekleri

| A.2.1 tarafından gönderilen ulaşılabilir adres ön ekleri | A.2.2 tarafından gönderilen ulaşılabilir adres ön ekleri | A.2.3 tarafından gönderilen ulaşılabilir adres ön ekleri |
|----------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------|
| P<A.2.1>                                                 | P<A.2.2>                                                 | P<A.2.3>                                                 |
| P<Y.2>                                                   | P<Y.1>                                                   |                                                          |
| P<W.1.1.1>                                               | P<Z.2>                                                   |                                                          |

Ulaşılabilir adres ön ekleri gibi ulaşılabilirlik bilgileride, PNNI yönlendirme hiyerarşisinden beslenirler, böylece tüm düğümlerin adresleri bu ön eklerle özetlenmiş olarak uç sistemlere ulaşabilirler. Bu bilgi akışını mümkün olduğu yerlerde daha fazla özetlemeyi amaçlayan bir filtre etme işlemi bulunmaktadır. Örneğin LGN A.2, A.2 peer grubunda tanınmış her ulaşılabilir adresi özetlemeyi amaçlar, bunu listesinde bulunan tüm özet adreslerle eşleştirerek yapar (Tablo 2.2). Örneğin LGN A.2, düğüm A.2.2 tarafından işlenen P<Y.1> ulaşılabilir adres ön ekini PGL A.2.3 yoluyla aldığı (Tablo 2.1) ve de kendi yapılandırılmış özet adresi P<Y>’ye uygun bir eş bulduğunda, LGN A.2 daha uzun bir ulaşılabilir adres ön eki olan P<Y.1> yerine P<Y> özet adresini tanıtarak daha iyi bir özetleme elde eder.

LGN A.2 tarafından daha fazla özetlenemeyen ulaşılabilir adres ön ekleri değiştirilmeden tanıtılırlar. Örneğin LGN A.2, A.2.2 tarafından şekillendirilen P<Z.2> ulaşılabilir ön ekini aldığı kendine ait özet adresleriyle (Tablo 2.2) eşleştirmede başarısız olur ise, sonuç olarak LGN A.2 P<Z.2>’yi değiştirmeden tanır. Burada önemli olan nokta, LGN A.2, P<Z.2>’yi, A.2.2 düğümünün perspektifinden P<Z.2> bir özet adres olsa da, yabancı adres olarak görmesidir. Bunun sebebi kendine ait özet adreslerle yaptığı eşleştirme denemesini başarısız olmasıdır. Sonuç olarak LGN A.2 tarafından tanınan ulaşılabilirlik bilgisi Tablo 2.4’de gösterilmiştir:

Tablo 2.4 Tanınan Ulaşılabilirlik Bilgisi

| LGN A.2 tarafından tanınan ulaşılabilirlik bilgisi |
|----------------------------------------------------|
| P<A.2>                                             |
| P<Y>                                               |
| P<Z.2>                                             |
| P<W.1.1.1>                                         |

Tablo 2.3 ve Tablo 2.4’de görüldüğü gibi A.2.3 tarafından tanınan ulaşılabilirlik bilgisi, A.2.3 düğümü A.2 peer grubunun PGL’si olsa da, LGN A.2 tarafından tanınanlardan farklıdır. LGN A.2 tarafından tanınan ulaşılabilirlik bilgisi, peer grup dışında bu peer grup hakkında mevcut tek ulaşılabilirlik bilgisidir.

LGN A ile A.2 peer grup lideri arasındaki ilişki, LGN A.2 ile A.2.3 peer grubu arasındaki ilişkinin bir benzeridir. LGN A özet adresler olmaksızın yapılandırılrsa da, A peer grubundan akan tüm ulaşılabilir adres ön eklerini en yüksek peer grubuna tanıtırlar (Tablo 2.4’deki listenin tamamı dahil). Diğer yandan LGN A, varsayılan özet adres P<A> olmaksızın yapılandırılırsa (varsayılandır çünkü A peer grubunun kimliği PG(A)’dır) adresleri tanıtmadan önce P<A> ile başlayan tüm ulaşılabilir adres ön eklerini daha da özetlemeye çalışacaktır. Örneğin, P<A.2> adres ön ekinin yerine P<A> özet adresini tanıtacaktır (Tablo 2.4).

Mantıksal düğümlerin ATM adresleri, uç sistem adresleri ile aynı özetleme kurallarına tabidir.

Belirli bir PNNI düğümü tarafından işlenen ulaşılabilirlik bilgisi (ulaşılabilir adres ön ekleri) grup boyunca ve yukarı doğru baba peer gruplarına tanıtılır. Daha sonra da grup boyunca ve aşağı doğru çocuk peer gruplarına tanıtılır, ve sonunda ulaşılabilir adres ön ekleri, bu belirli düğüm dışında bulunan tüm PNNI düğümlerine ulaşır.

### 2.6.1 Adres Kapsamı

Bir mantıksal düğüm tarafından tanıtılan ulaşılabilirlik bilgisinin her zaman bir kapsamı vardır. Kapsam, PNNI yönlendirme hiyerarşisinde bir seviyeyi belirtir ve bir adresin tanıtılacağı veya özetleneceği en üst seviyedir. Eğer adresin, düğümün seviyesinden daha düşük bir seviyeyi gösteren bir kapsamı varsa, düğüm adresi

tanıtmaz. Eğer kapsam, düğümün seviyesine eşit veya düğümün seviyesinden daha yüksek bir seviye gösteriyorsa, adres düğümün peer grubunda tanıtılır.

Adresler özetlenirken en yüksek kapsamlı özetlenecek adres, özet adresin kapsamını belirler. Aynı kural grup adresleri için de geçerlidir, örneğin bir peer grupta iki veya daha fazla düğüm aynı grup adresine fakat farklı kapsamla ulaşılabilirlik ilan ederse, baba düğümleri en yüksek kapsamlı grup adresine ulaşılabilirlik ilan eder.

## 2.7 Tek Düğüm Perspektifi

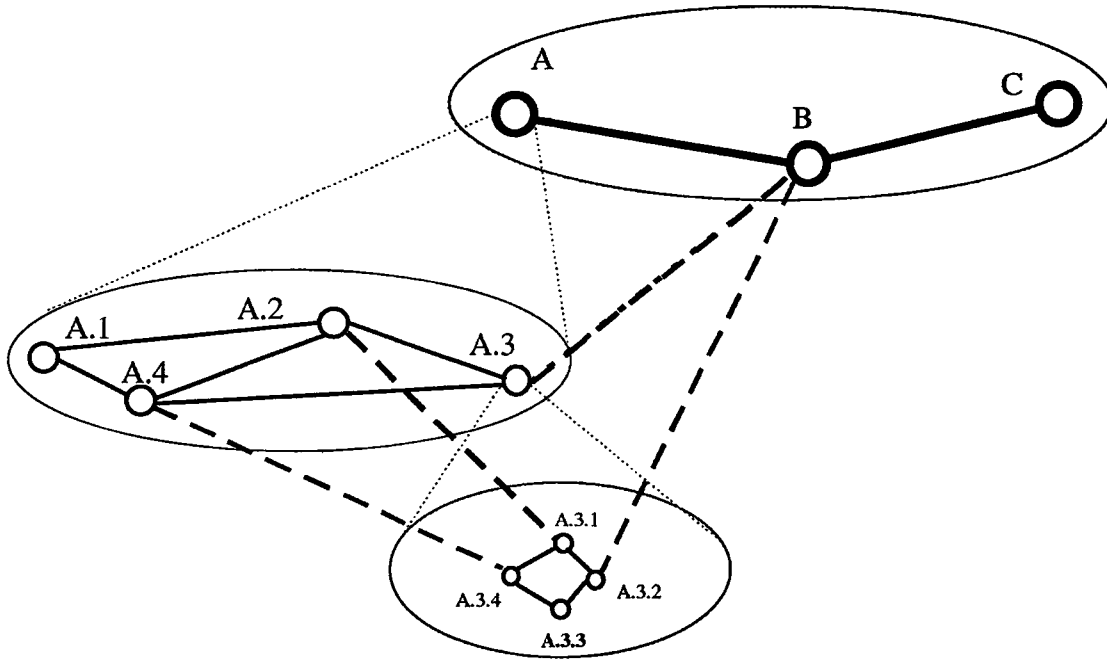
Bu bölümde, bir en düşük seviye düğümün ağ hakkında sahip olduğu perspektif ele alınmaktadır. Ağın görüntüsü düğümün topoloji veritabanında bulunan ağ hakkındaki bilgiler kullanılarak yaratılmaktadır.

Şekil 2.10'da, Şekil 2.6'da gösterilen PNNI yönlendirme hiyerarşisinin, A.3 en düşük seviye peer grubunda bulunan tüm düğümlerin sahip oldukları ağ görüntüsü verilmiştir. Bu ağ görüntüsü, A.3.1'den A.3.4'e kadar peer grupta bulunan 4 düğüm için de aynıdır. Bu ortak görüntü A.3 peer grubunda bulunan tüm düğümlerin bilgilerini, taşkın yöntemiyle birbirlerine göndererek topoloji veri tabanlarını aynı tutmalarıyla sağlanır.

Şekil 2.10'daki görünümde, A.3 peer grubunun tüm ata peer grupları bulunmaktadır, örneğin: A peer grubu ve en yüksek seviye peer grubu. Genellemek gerekirse, bir peer grubun ağın geri kalanı hakkında sahip olduğu görüntü, o grubun baba peer grubuna ve babanın kendi baba peer grubuna karşılık gelir. Bu durum herhangi bir hiyerarşik seviyedeki herhangi bir peer grup için geçerlidir.

Şekil 2.10'daki örnek, A.3.3 düğümü üzerinde odaklanmıştır ve bu düğümün çevresindeki düğümleri nasıl gördüğünü inceler. Senaryoyu basitleştirmek için A.3.3'ün komşularının A.3.3 oluşmadan önce aktif olduğu varsayılmıştır. A.3.3 çalışmaya başladığında iki yeni bağ belirir, (A.3.3- -A.3.4) ve (A.3.3- -A.3.2). Bu yeni bağların üzerindeki Hello protokolü A.3.3 düğümüne yeni komşularını bildirir; A.3.2 ve A.3.4.

A.3.3 düğümü artık komşularından başlangıç veritabanı değişimini ve taşkın yoluyla topoloji veritabanının tamamını elde edebilir.



Şekil 2.10 A.3.3, A.3.2, A.3.1 ve A.3.4 düğümlerinden ağın görüntüsü

A.3.3'ün tek düğüm perspektifi topoloji veritabanındaki PTSE'lerden yararlanarak oluşur. A.3.3 düğümü A.3 peer grubunun topolojisini, A.3 peer grubundaki peer'ların oluşturduğu PTSE'lerdeki bilgilerden öğrenir. A baba peer grubundaki A.4 üst düğümüne bağlanabilirlik (Şekil 2.6), A.3.4 sınır düğümü tarafından yaratılan ve aktarılan (A.3.4- -A.4) üst bağından öğrenilir. Benzer şekilde, A.2 üst düğümüne bağlanabilirlik, A.3.1 sınır düğümü tarafından yaratılan ve aktarılan (A.3.1- -A.2) üst bağından öğrenilir. A.3 peer grubundaki hiçbir düğümden A.1 düğümüne bir üst bağ yoktur. Üst bağlar sadece düşük seviye peer grupları ile komşu üst düğümler arasındaki bağlanabilirlik hakkında bilgi sağlarlar (örneğin A.4 ve A.2). A.3.3 düğümünün A.1, A.2 ve A.3 düğümleri ve aralarındaki bağlanabilirlik hakkındaki görüşü, bu düğümler tarafından oluşturulan PTSE'ler sayesinde öğrenilir.

Şekil 2.10'daki topoloji özeti, A.3 peer grubuna ait olmayan tüm düğümler için dolaylı şekilde verilmiştir. Örnekte bu durum, Şekil 2.6'daki (A.2.1- -A.4.1) ve (A.2.2- -A.4.2) bağlarının toplamını temsil eden (A.2- -A.4) bağı için gösterilmiştir.

Topoloji veritabanında A.3.3'ün, her düğümden ulaşılabilen adres ön eklerini anlatan ilanları olacaktır. Örneğin, A.2.1, A.2.2, A.2.3 (Şekil 2.6 ve Şekil 2.9) düğümlerine bağlı uç sistemlerle ilgili ulaşılabilir adres ön ekleri, A.2 mantıksal düğümünden gelen ilanlarda özetlenir.

## 2.8 Güzergah Seçimi

Güzergah seçimi, işaretleşme talebinin ATM ağı içerisinde yönlendirilmesi için kullanılır. Bu yönlendirmeyle oluşturulan güzergah, ATM bağlantısının kurulduğu ve verilerin akacağı güzergahtır. Bir ATM ağında işaretleşme talebinin yönlendirilmesi, ATM'in bağlantı tabanlı yapısına rağmen, diğer ağ katmanı protokollerindeki (IP gibi) bağlantısız paket yönlendirmesine benzerlik göstermektedir[6]. Bunu doğuran sebep ise bağlantı kurulmasından önce, işaretleşme talebinin takip edeceği bir bağlantının bulunmamasıdır.

PNNI yönlendirme protokolü varolan yönlendirme protokollerinden çok daha karmaşıktır. Bunun sebebi, bu protokolün büyük çaplı ağların ölçeklenerek desteklenebilmesi ve QoS ile yönlendirmenin sağlanmasını amaçlamasıdır.

ATM'nin büyük avantajlarından bir tanesi bağlantılarda sunduğu QoS garantisidir. Bir düğüm bağlantı talebinde bulunurken bu bağlantı için belli QoS'lar ister. Bağlantılar, sağladıkları QoS tiplerine göre kategorize edilirler. Bunlar, arzu edilen QoS garantisine ve beklenen trafiğin karakteristiğine göre CBR, VBR, ABR ve UBR olarak ayrılır. Ağ, düğümün QoS taleplerini tüm bağlantı boyunca karşılayabilmelidir[7].

### 2.8.1 Bağlantı Kabul Denetimi

QoS garantisinin sağlanması için düğümler bağlantı kabul denetimi (CAC) adlı bir fonksiyonu kullanırlar. Düğüm bağlantı talebini aldığı zaman, CAC fonksiyonunu gerçekleştirir. Bu fonksiyon, bağlantının talep edilen QoS ve trafik parametrelerini baz alır ve kurulmak istenen bağlantının QoS garantisini sağlayıp sağlayamayacağını kontrol eder. CAC yerel bir düğüm fonksiyonudur ve yerel denetim kurallarının QoS garantisi üzerindeki kesinliğine bağlıdır.

### 2.8.2 Genel Bağlantı Kabul Denetimi

Yönlendirme protokolü, bağlantı talebinin hedefe giden güzergah üzerinden yönlendirilmesini ve bağlantı kurulumu esnasında talep edilen QoS metriklerinin yüksek olasılıkla sağlanmasını garanti altına almalıdır. Fakat bu, PNNI gibi, hiyerarşik yapıya sahip ağlarda oldukça zordur. Çünkü hiyerarşik yapı, ulaşılabilirlik bilgisinin özetlenmesini, bağ ve düğüm parametrelerinin de birleştirilmesini

gerektirir. Yapılan birleştirme işlemi kayıplı bir işlemdir ve yanlışlıklara yol açmaktadır. Bunlara ek olarak yukarıda da belirtildiği gibi CAC yerel bir fonksiyondur ve kullanılan yerel politikalara bağlıdır.

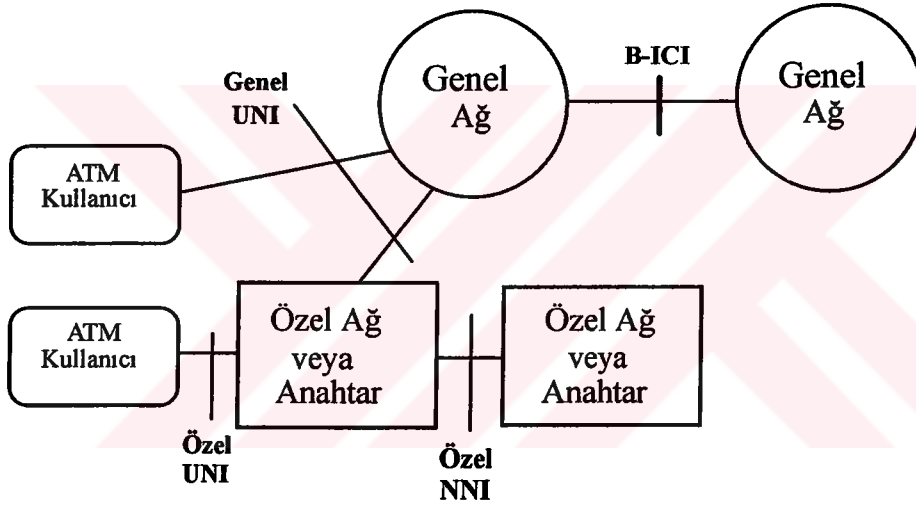
PNNI protokolü bu problemi genel bağlantı kabul denetimi (GCAC) fonksiyonunu tanımlayarak kontrol altına alır.



### 3 PNNI İŞARETLEŞMESİ

PNNI protokolünün ikinci birleşeni bir işaretleme protokolüdür. İşaretleme protokolleri iletişim ağlarında çağrı ve bağlantı ayarları için kullanılır. Bu protokoller çağrıları oluşturulmasında ve kontrolünde gerekli bilgileri değiş tokuş eder. Temel olarak ATM işaretlemesi, bir ATM ağında ATM arayüzleri arasında çağrı/bağlantı kurulması ve bunların kaldırılması için kullanılan bir dizi protokoldür.

Şekil 3.1’de ATM ağlarının desteklediği arayüzler görülmektedir.



Şekil 3.1 ATM arayüzleri

**Genel UNI :** Bir ATM kullanıcısı ve bir genel ATM ağı arasındaki kullanıcı-ağ arayüzü

**Özel UNI :** Bir ATM kullanıcısı ve bir özel ATM ağı arasındaki kullanıcı-ağ arayüzü

**B-ICI :** İki genel ağ veya anahtar sistem arasındaki ağ-ağ arayüzü. B-ICI, BISDN Inter Carrier Interface’in kısaltılmışıdır.

**Özel-NNI :** İki özel ağ veya anahtar sistemi arasındaki ağ-ağ arayüzü

PNNI işaretleşmesi, UNI 4.0 işaretleşmesi baz alınarak geliştirilmiştir[8]. Bazı UNI 4.0 işaretleşme özelliklerini desteklememekle beraber, dinamik çağrı kuruluşu için PNNI yönlendirme kullanımına ilişkin bazı yeni özellikler de eklenmiştir.

İşaretleşme, bağlantı kurmak isteyen bir uç-sistem tarafından başlatılır. Kaynak uç sistem hedef uç sistemin adresini ve arzu ettiği trafik ve QoS parametrelerini içinde bulunduran bir SETUP mesajı yaratır. Bu mesaj UNI aracılığıyla, ağdaki ilk anahtara ulaşır. Kaynak ve hedef UNI arasında, ATM bağlantı talebini ağ içinde aktarmada kullanılan PNNI işaretleşme protokolü devreye girer. UNI işaretleşme talebi, kaynak anahtarda, NNI işaretleşmesine eşlenir. Bağlantı mesajı, ağ boyunca, hedef uç sisteme ulaşana dek anahtardan anahtara ilerler. PNNI işaretleşmesi ise, hedef anahtarda, tekrar UNI işaretleşmesine eşlenir. Bağlantı isteğinin yönlendirilmesi ve takip eden tüm veri akışı PNNI yönlendirme protokolü tarafından yönlendirilir. PNNI protokolü bu bağlantı talebini varış adresine, ve kaynak tarafından istenen trafik ve QoS parametrelerine göre yönlendirir. Hedef uç sistem ya bağlantı talebini kabul edip onaylar ya da reddedip bağlantıyı temizler. Bağlantı, bağlantı talebinin izlediği güzergah üzerinde kurulduğu için veriler de aynı güzergahı takip ederler.

PNNI işaretleşmesi, PNNI yönlendirmesi tarafından ağdan toplanan bilgilerden yararlanır. Özellikle, PNNI yönlendirmesi tarafından toplanan ulaşılabilirlik, bağlanabilirlik ve öz kaynak bilgilerinden hesaplanan güzergah bilgilerini kullanır.

Gereksinimleri karşılamak amacıyla , PNNI işaretleşmesine, UNI işaretleşmesi için yapılan tanımlara yeni özellikler eklenmiştir. Bunlar :

Belirlenmiş Geçiş Listesi (DTL), hiyerarşik olarak tanımlanmış kaynaktan yönlendirme bilgilerini taşımak amacıyla kullanılır.

Crankback ve çok seçenekli yönlendirme, ağ hakkında yetersiz ve ağın gerçek durumunu kesin olarak yansıtamayan bilgilerle başa çıkmakta ve alternatif güzergahlar belirlemede kullanılır.

Ortak işaretleşme, PNNI'ın sanal güzergah bağlantıları üzerinde çalışması için kullanılır.

### 3.1 Ortak İşaretleşme

Varsayılan olarak (eğer hiçbir VPC, arayüzde, mantıksal bağ olarak kullanılmak üzere konfigüre edilmemişse), fiziksel arayüzdeki tüm sanal güzergahlar,  $VP=0$  işaretleşme kanalından kontrol edilir. Ayrıca PNNI, bir çok sanal güzergah bağlantısının (VPC'ler) bir çok hedefe tek bir fiziksel arayüz üzerinden işaretleşme ve yönlendirilmesini destekler. Bu durumda, mantıksal bağ olarak kullanılmak üzere yapılandırılmış her VPC'nin ortak işaretleşme kanalı olur. Bu VPC'ler içindeki sanal kanallar o VPC'nin işaret kanalı ile kontrol edilirler, örneğin aynı arayüzdeki varsayılan işaretleşme kanalı ( $VPI=0$ ) VPC'ler içindeki, mantıksal bağ olarak kullanılan, sanal kanalları kontrol etmez sadece fiziksel bağda geri kalan tüm sanal kanalları ve sanal güzergahları denetler.

### 3.2 Belirlenmiş Geçiş Listesi

Bir çağrıyı işlerken PNNI işaretleşmesi, PNNI yönlendirmesinden bir güzergah belirlemesini talep edebilir. PNNI kullandığı güzergahları DTL kullanarak belirtir. DTL, bir dizi düğüm kimliğinden ve isteğe bağlı olarak kullanılan peer grupların kapı kimliklerinden oluşan, bir peer grup içindeki güzergahı tam olarak belirten bir geçiş listesidir. DTL, kaynak düğüm tarafından (örn. DTL yaratıcısı) veya bir peer gruba giriş sınır düğümü tarafından oluşturulur. Hiyerarşik olarak yapılandırılmış kaynaktan yönlendirme güzergahı, hem kaynağın hem de hedefin ulaşılabilir olduğu, görünen en düşük seviye ile o anki seviye arasındaki her hiyerarşik yönlendirme seviyesini içeren, bir güzergahtır. Bu yapı, en düşüktən en yüksek peer grup seviyesine doğru sıralanmış ve yığın şeklinde organize olmuş DTL'ler dizisi olarak ifade edilmiştir (son giren ilk çıkar veri yapısı). Yığının en üstünde yer alan DTL, en düşük seviye peer grubuna karşılık gelir.

Bir çağrının güzergahı boyunca her düğümde gerçekleştirilen DTL işlem prosedürleri iki bölümden oluşur: İlk bölümde, düğüm, DTL yığınının herhangi bir DTL'nin eklenip eklenmemesine karar verip gereken önlemleri alır; ikinci bölümde ise, düğüm, yığından herhangi bir DTL'nin kaldırılıp kaldırılmaması gerektiğine karar verir ve kalan DTL yığınının sonraki düğüme göndermek üzere işler. Tipik olarak, bir düğüm ya DTL yığınının ekleme yapar ya da buradan DTL kaldırır. Eğer bir düğüm DTL'leri hem ekleyip hem kaldırıyorsa, ilk olarak eklediği DTL'yi kaldırmalıdır.

### 3.3 Crankback

Bir DTL yaratırken düğüm, o anda veritabanında bulunan ağa ait öz kaynak ve bağlanabilirlik bilgilerini kullanır. Ağ hakkındaki bu bilgiler bazı sebeplere bağlı olarak yetersiz veya ağın gerçek durumunu yansıtmıyor olabilirler. Bu durum temel olarak, yönlendirme bilgisinin üretildiği andan itibaren bağlantının kurulmasına kadar ki sürede kurulumu tamamlanan ek bağlantılar nedeniyle düğümlerin sunabildiği öz kaynaklarda oluşan değişiklikler ve bilgilerin, kurulan hiyerarşi gereği ağda özetlenerek dağıtılmasıdır. Bu yüzden bir DTL'ye göre kurulmaya çalışılan bir çağrı, belirlenmiş güzergahı boyunca bloke olabilir. Crankback ve çok seçenekli yönlendirme, çağrıyı ilk kaynağına geri göndermek yerine, mümkünse bu duruma adapte olarak çağrıyı, kısmen kaldıran ve yeniden yönlendirmeyi deneyen bir mekanizmadır. Çağrı belirlenen DTL'ye göre kurulamadığı zaman o DTL'nin yaratıcısına sorunun nedeni de eklenerek crankback edilir. Bu düğüm çağrıyı yönlendirebileceği alternatif bir güzergah seçer ya da o da çağrıyı crankback eder. Belirlenecek alternatif güzergah, daha önce alınan tüm üst seviye DTL'lere uymalı ve bloke düğüm ve bağlardan kaçınmalıdır.

### 3.4 Crankback ile Bağlantı Kuruluşuna Bir Örnek

Bu bölümde, crankback ve çok seçenekli yönlendirmeyi içeren detaylı bir bağlantı kurulumu örneği verilmiştir. Örnekte kullanılan ağ Şekil 3.2'de gösterilmiştir. A.3.3 ve B.1.2 arasındaki bağın bloke olduğu (PNNI yönlendirme bilgisi sorunsuz olduğunu söylese de) ve A.3.1 ve B.2.3 arasındaki bağın çağrı tarafından talep edilen parametreleri destekleyemediği varsayılmaktadır.

Şekil 3.2' de gösterilen A.[1- -3] ve B.[1- -3] peer gruplarında, A.1.2 anahtar sistemine bağlı A.1.2.x uç sisteminden B.3.3 anahtar sistemine bağlı B.3.3.y uç sistemine bir bağlantının kurulacağını varsayalım. A.1.2.x, bağlantı isteğini normal bir UNI SETUP olarak A.1.2'ye gönderir. A.1.2 elindeki verilerle oluşturduğu ağ görüntüsünü inceler. B.3.3.y hedef adresinin B düğümü üzerinden ulaşılabilir olduğunu belirler. Peer grubu ismindeki B'nin, hedefin ATM uç nokta adresindeki B'den farklı olabileceğine dikkat edilmelidir. B'deki ilanın, ulaşılabilir adresin ön eklerini açıkça içermesinin sebebi budur.



DTL:[A.1, A.2, A.3], işaretçi-1

DTL:[A, B], işaret-1

Her DTL, verilen hiyerarşik seviyede çağrı kuruluşunun uğraması gereken düğümleri listeler (o anki düğüm de dahil olmak üzere). Her DTL'yi takip eden geçerli geçiş işaretçisi, o seviyede listedeki elemanlardan hangisinin ziyaret edilmekte olduğunu gösterir. Geçiş işaretçisi bir indeks olarak kodlanmamıştır, daha çok bir sekizli ofset şeklindedir. Kolay anlaşılabilmesi için indeks olarak gösterilmiştir. DTL listeleri bir DTL yığını olarak sıralanır (örneğin, en üst DTL harekete geçer; en üst DTL'nin sonuna ulaşıldığında bu DTL çağrı istek mesajından kaldırılır ve yığında yer alan bir sonraki DTL incelenir).

Çağrı kuruluşu gönderilmeden önce, çok seçenekli yönlendirmenin gerekmesi ihtimaline karşı A.1.2, SETUP mesajının içeriğini depolar. Daha sonra A.1.2 çağrı kuruluşunu komşusu A.1.1'e gönderir. A.1.1 en üst DTL'yi inceler ve DTL'nin kendi düğüm kimliğini işaret ettiğini fark eder. A.1.1 en üst DTL'de bir sonraki elemanı arar ve olmadığını görür. Bir sonraki DTL'ye geçer, bu DTL'deki bir sonraki hedef A.2'dir. A.1.1, A.2 'ye ait olmadığından (örneğin, A.1.1, A.2 mantıksal düğümü içine özetlenmiş, peer grubuna ait değildir), A.2'ye ulaşmak için bir yolu aramaya başlar. A.1.1, A.2'de bir komşusunun olduğunu görür, böylece yığından en üst DTL'yi kaldırır ve bir sonraki DTL'deki geçerli geçiş işaretçisini artırır:

DTL: [A.1, A.2, A.3] işaretçi-2

DTL: [A,B] işaretçi-1

A.1.1, yığına bir DTL eklememiş ve herhangi bir yönlendirme kararı vermemiştir, böylece A.1.1 çok seçenekli yönlendirme işlemine karışmaz ve A.2'ye çağrı kurma mesajını göndermeden önce SETUP mesajının bir kopyasını almaya ihtiyaç duymaz.

A.2.2 en üst DTL'ye bakar, ve geçerli hedefin A.2 olduğunu görür. A.2.2, A.2'de olduğundan DTL'deki bir sonraki elemana bakar ve A.3'e doğru yönlendirmeye başlar. Topolojiyi analiz ederek A.2.2, A.3'e giden güzergahın A.2.3'den geçtiğini bulur ve o DTL'yi yığının üstüne ekler:

DTL: [A.2.2, A.2.3] işaretçi-2

DTL: [A.1, A.2, A.3] işaretçi-2

DTL: [A, B] işaretçi-1

A.2.2, yığına bir DTL eklediğinden ve çok seçenekli yönlendirme işlemine dahil olabileceğinden aldığı SETUP mesajının içeriğini, yığının üstüne eklediği DTL'nin içeriğini ve çağrı paketini A.2.3'e göndermeden önce depolar.

A.2.3 ilk olarak en üst DTL hedefine ulaşıldığını ve sonra DTL'nin tükendiğini belirler. Son olarak A.2.3 bir sonraki DTL'deki diğer hedefin komşu olduğunu fark eder ve böylece A.2.3 en üst DTL'yi kaldırıp geçerli geçiş işaretçisini bir sonraki DTL'ye iletir:

DTL: [A.1, A.2, A.3] işaretçi-3

DTL: [A, B] işaretçi-1

Çağrı kurma mesajı A.3.4'e ulaşır. A.3.4, A.3'ün içinde olduğundan hedefe ulaşılmıştır. Hedefi B olarak terk eder. A.3.4, A.3.2 ve A.3.3 üzerinden B'ye bir güzergah oluşturur ve yığına yeni bir DTL ekler:

DTL: [A.3.4, A.3.2, A.3.3] işaretçi-2

DTL: [A.1, A.2, A.3] işaretçi-3

DTL: [A, B] işaretçi-1

A.3.4 yığına yeni bir DTL eklediğinden ve çok seçenekli yönlendirmeye dahil olabileceğinden A.3.4, çağrıyı A.3.2'ye iletmeden önce SETUP mesajının içeriğini depolar.

A.3.2 çağrı kuruluşunu alır, geçerli geçiş işaretçisini arttırır, ve çağrı kurma isteğini A.3.3'e yöneltir. A.3.3., çok seçenekli yönlendirmeye dahil olmayacaktır ve bu yüzden SETUP mesajının kopyasını saklamaya gerek duymaz. A.3.3 çağrı kurma mesajını alır ve SETUP mesajını B'deki komşusuna iletmeye karar verir. SETUP mesajını iletmeden önce A.3.3, en üstteki iki DTL'yi kaldırır; çünkü her ikisi de tükenmiştir ve SETUP mesajı A.3.3 ve A.3'de ayrılmaktadır:

## DTL: [A, B] işaretçi-2

Bu noktada, çağrı kuruluşu bloke olmuştur, çünkü A.3.3 (önceki düğüm), bir Crankback Bilgi Elemanı (IE) içeren ya CAC dahiliden A.3.3'e, ya da bir RELEASE veya RELEASE COMPLETE mesajından dolayı, bu arayüzün sonraki ucunda çağrıyı bloke olmuş.bulur (Şekil 3.3'de 1 etiketli daire tarafından belirtildiği gibi). A.3.3'ün önceki arayüze bir RELEASE mesajını geri göndermesiyle Crankback prosedürü başlamıştır. RELEASE mesajındaki Crankback IE, çağrının, SETUP mesajından alınan DTL'lerden belirlendiği şekilde A.3.3 ve B arasındaki bağda bloke olduğunu ve Crankbank seviyesinin 96 olduğunu belirtir (A.3 peer grubunun seviyesi).

RELEASE mesajı bu çağrı için bir DTL yaratmamış olan A.3.2'de alınır, bu yüzden crankback ilerlemeye devam eder. Takip eden sanal kanalın yıkılmasından sonra veya yıkılması sırasında RELEASE mesajı önceki arayüze A.3.4 yönünde gönderilir.

RELEASE mesajı seviye 96'daki çağrısı için bir DTL belirleyen A.3.4'de alınmıştır, böylece A.3.4 çok seçenekli yönlendirme yapmayı dener. Bloke olmuş bağ elendikten sonra, A.3 ile B arasında iki farklı güzergah bulunur. Bir seçenek A.3.1 yoluyla B üzerine, diğer seçenek ise A.3.2, A.3.3, A.3. yoluyla B üzerinedir. Ama, A.3.1 ve B arasındaki bağ, orijinal SETUP mesajında talep edilen yüksek hücre oranını karşılayamaz, bu yüzden A.3.4 kalan seçeneklerden birini denemek yerine crankback prosedürüne devam etmeye karar verir çünkü orijinal güzergahta daha iyi güzergahlar bulunabilir. A.3.4, önceki arayüze bir RELEASE mesajı daha gönderir, ama A.3 ile B arasındaki bağın bloke olduğunu göstermek için bloke olmuş geçiş belirtecini değiştirir. Ayrıca, 2 etiketli daire ile belirtildiği gibi Crankback seviyesini 72 olarak değiştirir (A.1'in peer grubunun seviyesi).

RELEASE mesajı A.2.3'de alınır. A.2.3 bu çağrı için hiçbir DTL üretmediğinden crankback A.2.2 yönünde ilerlemeye devam eder.

RELEASE mesajı A.2.2'de alınır. A.2.2 bu çağrı için bir DTL üretmiştir. A.2.2 IE crankback'inde crankback seviyesini inceler ve seviyenin yaratılan DTL'nin seviyesinden daha yüksek olduğunu belirler, bu yüzden crankback A.1 yönünde ilerlemeye devam eder.

RELEASE mesajı A.1.1’de alınır. A.1.1 bu çağrı için hiçbir DTL üretmediğinden crankback A.1.2 yönünde ilerlemeye devam eder.

RELEASE mesajı A.1.2’de alınır. A.1.2, crankback seviyesinin bu çağrı için belirlediği DTL’in seviyesiyle uyuştuğunu görür ve bu uyuşan SETUP mesajının içeriğinin bir kopyasını bulur. Çok seçenekli yönlendirme, 3 numaralı dairenin belirttiği gibi denenir, ve A.1.2 (A.1.2, A.1.1, A.2, B) güzergahını dener. Bunun sonucu olarak aşağıdaki DTL yığını ortaya çıkar:

DTL: [A.1.2, A.1.1], işaretçi-2

DTL: [A.1, A.2], işaretçi-1

DTL: [A, B], işaretçi-1

Daha sonra A.1.2 çağrı kuruluşunu A.1.1’e gönderir. A.1.1, en üst DTL’yi yığından kaldırır, geçerli geçiş işaretçisini bir sonraki DTL’ye aktarır ve çağrıyı A.2.2’ye gönderir. A.2.2, A.2.1’den B’ye giden bir güzergah bulur, yeni DTL’yi yığına ekler ve çağrı kurma mesajını A.2.1’e gönderir:

DTL: [A.2.2, A.2.1], işaretçi-2

DTL: [A.1, A.2], işaretçi-2

DTL: [A, B], işaretçi-1

A.2.1 çağrı kurma mesajını alır ve B’nin komşu olduğunu fark eder. En üst iki DTL’yi yığından çıkarır, bir sonraki DTL’deki geçerli geçiş işaretçisini arttırır ve Şekil 3.3’deki 4 numaralı dairenin belirttiği gibi çağrı kurma mesajını B.1.1’e gönderir:

DTL: [A, B], işaretçi-2

B.1.1 çağrı kurma mesajını alır ve geçerli DTL hedefine ulaşıldığını görür. B.1.1 son hedefi oluşturmak için artık yeni bir güzergah geliştirmelidir. Güzergahın B.2’den B.3’e gideceğini fark eder ve onu iter. Sonra B1’den giden güzergahın B.1.3’e giden olduğunu fark eder ve yığına ekler:

DTL: [B.1.1, B.1.3], işaretçi-2

DTL: [B.1, B.2, B.3], işaretçi-1

DTL: [A, B], işaretçi-2

Daha sonra B.1.1 çağrı kurma mesajını B.1.3'e gönderir. B.1.3, her zamanki gibi denetledikten sonra, en üst DTL'yi kaldırır, DTL işaretçisini arttırır ve B.2'deki komşusuna verir. B.2.2 geçerli hedef olduğunu fark eder ve B.2'den B.3'e bir güzergah bulduktan sonra onu yığına ekler:

DTL: [B.2.2, B.2.1, B.2.2.3], işaretçi-2

DTL: [AB1, B.2, B.3], işaretçi-2

DTL: [A, B], işaretçi-2

Daha sonra B.2.2 çağrı kurma mesajını B.2.1'e gönderir.

B.2.1 en üst DTL'ye bakar, onu ilerletir, ve çağrı kurma mesajını B.2.3'e gönderir. B.2.3 DTL'yi kaldırır, işaretçiyi ilerletir ve çağrı kurma mesajını B.3'deki komşusuna gönderir. B.3.4 yeni bir DTL yaratır:

DTL: [B.3.4, B.3.1, B.3.3], işaretçi-2

DTL: [B.1, B.2, B.3], işaretçi-3

DTL: [A, B], işaretçi-2

B.3.1 en üst DTL'ye bakar, geçerli geçiş işaretçisini ilerletir, ve çağrı kurma mesajını B.3.3'e gönderir. B.3.3, bunun DTL yığını için DTL sonlandırıcısı olduğuna karar verir, çünkü:

- Tüm DTL'lerin en sonuna ulaşılmıştır,
- B.3.3, bir en düşük seviye düğümdür,
- B.3.3, hedefe ulaşabilir durumdadır.

Daha sonra B.3.3, DTL yığınıni SETUP mesajından çıkarır ve bu mesajı UNI üzerinden hedefe gönderir.

Özetle, DTL işlenmesinde kullanılan algoritma şu adımlardan oluşur:

- Geçerli düğümün belirlenmesi (geçerli hiyerarşi seviyesi dahil olmak üzere)
- Güzergah seçimi hedefinin belirlenmesi
- Güzergahın belirlenmesi ve güzergahın DTL yığına eklenmesi
- Tamamlanmış DTL'lerin yığından çıkarılması
- Geçerli geçiş işaretçisinin ilerletilmesi

### 3.5 Crankback Oluşmasının Sebepleri

PNNI'da crankback mekanizmasının kullanımını zorunlu kılan durumlar üç ana kategoride toplanabilirler:

- Ulaşılabilirlik hataları
- Kaynak hataları
- DTL işleme hataları

#### 3.5.1 Ulaşılabilirlik Hataları

Genel olarak ulaşılabilirlik hataları, hedefe giden bir güzergahın bulunamadığını belirtir. Buna sebep güzergah üzerindeki bir bağı veya düğümün SETUP mesajı tarafından geçilmeden önce devre dışı kalması ya da çökmesidir. Böylesine bir durum, hem SETUP mesajının içindeki DTL'lerdeki ara hedefler için hem de son hedef için geçerli olabilir. Bu tip bir hata, sadece alınan DTL yığınının kapsamı içinde hedefe giden bir güzergahın olmadığı anlamına gelir ve kesinlikle, hedefe giden bir güzergahın bulunmasından fakat bunun bağlantının talep ettiği QoS'u karşılamamasından farklıdır.

#### 3.5.2 Kaynak Hataları

Bir çağrıyı desteklemek için gereken kaynaklar, SETUP mesajında bulunan trafik parametreleri ve QoS parametreleri kullanılarak hesaplanır. Kaynak hataları, çağrının

talep edilen trafik ve QoS parametrelerini karşılayabilen bir güzergahın bulunamadığını belirtmek için kullanılır.

Buna sebep olarak, toplanabilir bir metriğin değerinin (örneğin gecikme veya maliyet gibi) çağrının talep ettiği maksimum değeri aşması ya da bir bağa veya peer gruba ait toplanabilir olmayan bir metriğin (örneğin bant genişliği gibi) çağrının talep ettiği değeri karşılayacak kadar büyük olmaması gösterilebilir.

Talep edilen trafik ve QoS parametrelerinin karşılanamayacağı ya GCAC/güzergah seçimi sırasında, ya da CAC sırasında fark edilir. Yetersiz kaynaklar sebebiyle reddedilen çağrılar her zaman PNNI tarafından crankback edilir.

Eğer ATM trafik tanımlayıcı elemanından talep edilen kullanıcı hücre oranları karşılanamazsa, çağrı “yeterli kullanıcı hücre oranı mevcut değil” mesajı ile crankback edilir. Ayrıca, bloke olmuş düğüm veya bağ ile ilgili genel CAC parametrelerinin güncellenmiş değerleri de bu mesaja dahil edilebilir.

### **3.5.3 DTL İşlenme Hataları**

Düğüm, içinde geçiş işaretçisi tarafından belirtilen mantıksal düğüm belirtecinin bir düğümü veya atalarından birini tanımlamadığı bir belirlenmiş geçiş listesi bilgi elemanı aldığında, crankback meydana gelebilir. Bu tür DTL işlem hataları bağlanabilirlik değişimleri sonucunda geçici olarak ortaya çıkabilirler, bu durum özellikle peer grup lideri değiştiğinde ve yeni bir baba peer grubu yaratıldığında oluşur. DTL işleme hatalarına, ulaşılabilirlik hatalarının alt kümesi olarak bakılabilir, çünkü bunlar genelde bağlanabilirlikte oluşan değişikliklerin sonucudur.

Crankback, bir bağlantı kurulumunda hata oluşması durumunda, bağlantının, güzergah üzerindeki bir ara düğüm tarafından geri döndürülmesidir. Bu ara düğüm hedefe ulaşmak için farklı bir güzergah bulmayı dener. Bunu yaparken de başlangıç düğümünün işlemlerini uygular fakat daha güncel ve daha doğru ağ bilgilerini kullanır. Crankback oluşumundan sonra yapılacak işlemler crankback'in oluşum sebebine bağlıdır. Bu duruma ilişkin hata kodu bir evvelki adıma döndürülen mesaj içinde yer alır. Her ne kadar crankback oluşumundan sonra uygulanması gereken prosedürlerin ana hatları verilmişse de, bunların gerçekleşmesindeki ara adımlar

belirtilmemiş ve bazı tasarım kararları uygulamayı gerçekleştirecek firmaların inisiyatifine bırakılmıştır.

Crankback sağladığı faydalar yanında ağ üzerinde ek yükler de yaratır. Bu ek yük ise, çok fazla hiyerarşik seviyeye sahip geniş çaplı ağlarda, ağ hakkındaki QoS bilgileri doğru olarak özetlenemediği için, önemli boyutlara ulaşabilir. Bu ek yük temel olarak; PNNI bağlantı kurulumundaki gecikme, işaretleşme mesajlarının işlenmesi ve bu mesajların taşınmasıyla oluşan ek haberleşme yükü ve toplam VC kapasitesinin işaretleşmenin gerçekleşmesi için boşa harcanan bölümünden oluşur. Crankback'ın oluşturduğu ek yükünün azaltılması konusunda araştırmalar halen devam etmektedir. Bu çalışmalarda crankback'ın önceden kestirilmesi yönünde çeşitli öz kaynak kotalarının kullanılması ve bazı QoS parametrelerinin belirli değerleri aşmadan önce crankback mekanizmasının devreye sokulması amaçlanmaktadır[9]. Bu yaklaşımlar, başarılı sonuçlara ulaşmakla beraber, uygulanabilmek için PNNI standartına yeni eklentilerin yapılmasına ihtiyaç duyarlar.

## 4 PNNI YÖNLENDİRMESİNDE TEMEL MEKANİZMALAR

Bu bölümde PNNI yönlendirme protokolünün bünyesinde barındırdığı temel mekanizmalar ve protokoller detaylı bir şekilde incelenmektedir.

### 4.1 Hello Protokolü

PNNI Hello paketleri komşu düğümler arasında değişir. Bu değişim, RCC'ler üzerinden yapılır. Hellolar, komşu düğümlerin kimliğini doğrulamak ve bağların bu düğümlere olan durumunu belirlemek için tüm fiziksel bağlar ve VPC'ler üzerinden gönderilirler.

#### 4.1.1 Hello Durum Makinesi

En düşük seviyeli iki komşu düğüm arasında her fiziksel bağ veya VPC'nin kendi hello protokolü örneği vardır. Hello protokolünün örneği hello veri yapısı ve Hello durum makinesi tarafından yapılır.

Aralarında paralel fiziksel bağ ve/veya VPC'ler olan en düşük seviye düğümler için Hello protokolünün birden fazla örneği olacaktır. Ama, veritabanı zamanuyumlaması ve PTSE'lerin taşkını açısından komşu peer veri yapısı ve ortak komşu peer durum makinesinin tek bir örneği vardır.

##### 4.1.1.1 Hello Veri Yapısı

Düğümün tüm fiziksel kapıları ve tüm mantıksal kapıları için tek bir hello veri yapısı vardır.

Her bir hello veri yapısı aşağıdakilerden oluşur:

**Durum :** Bağın işlevsel durumu.

**Kapı Kimliği :** Düğüm tarafından atanan sayı; bu sayı, hello veri yapısı tarafından anlatılan fiziksel kapıyı veya, varsa, gerçek güzergah bağlantısını tanımlar.

**Uzak D ğ m Kimlięi :** Baęın dięer ucundaki komşu d ğ m n D ğ m Kimlięi.  
**Uzak D ğ m Kimlięi,** komşudan Hellolar alındıęında  ğrenilir.

**Uzak Peer Grup Kimlięi :** Baęın dięer ucundaki komşu d ğ m n Peer Grup Kimlięi.  
**Uzak Peer Grup Kimlięi,** komşudan Hellolar alındıęında  ğrenilir.

**Uzak Kapı Kimlięi :** Bir baę iin komşunun kapı kimlięi. Uzak Kapı Kimlięi, komşudan Hello paketleri alındıęında  ğrenilir. Uzak kapı kimlięi bilinmedięinde ise deęeri sıfıra ayarlanır.

**Hello Aralıęı :** Helloları tetikleyen olayların oluřmaması durumunda, d ğ m n bir baę  zerinden Helloları dıřarı g nderme aralıęıdır.

**Hello Zamanlayıcı :** Her Hello Aralıęında tetiklenen zamanlayıcıdır. Zamanlayıcının her tetikleyiřinde d ğ m bu baę  zerinden bir Hello yollar.

**Eylemsizlik Fakt r  :** Komşunun hellosu hen z ulařamadıęında d ğ m n baęı, kopmuř olarak varsaymasından  nce geen zamandır.

**Eylemsizlik Zamanlayıcısı :** Tetiklenmesi, bu komşudan yakın zamanda bir Hello almadıęı anlamına gelen zamanlayıcı. Eylemsizlik Zamanlayıcısının bařlangı deęeri, Eylemsizlik fakt r  arpı komşu d ğ mden en son alınan Hello Aralıęına ayarlanmalıdır.

**S r m :** Komşusu ile iletiřim kurma amalı olarak kullanılan Hello protokol n n s r m d r. Eęer kabul edilebilir bir s r m numarası t retilmemiřse alan sıfır olacaktır.

Dıř baęların durumunun belirlenebilmesi iin aynı peer grup iinde yer alan komşu d ğ mlerinkinden daha fazla bilgi gerekmektedir. Bu nedenle ( rneęin, dięer peer gruplardaki d ğ mlere olan baęlarda) ařaęıdaki ek alanlar da kullanılan veri yapısına dahil edilmiřtir:

**G nderilen  st Baę Bilgi  zellięi (ULIA) Sıra Numarası :** En son g nderilen ULIA'dan alınan sıra numarası.

**Alınan ULIA Sıra Numarası :** En son teslim alınan ULIA'dan alınan veya eęer alınan hello ULIA iermiyorsa, silinen sıra numarası.

**Alınan D ğ msel Hiyerar i Sıra Numarası :** En son alınan d ğ msel hiyerar i listesinden alınan veya e er en son alınan hello d ğ msel hiyerar i listesi i ermiyorsa, silinen sıra numarası.

** st D ğ m n D ğ m Kimli i :**  st d ğ m n D ğ m Kimli i, yeterli d ğ msel hiyerar i listesi olan Hellolar kom u d ğ mden alındı ında  ğrenilir.

**Ortak Peer Grup Kimli i :** Sınır d ğ m n veya dı  kom u d ğ m n n ortak peer grubunun peer grup kimli i. Ortak Peer Grup kimli i, yeterli d ğ msel hiyerar i listesi olan Hellolar kom u d ğ mden alındı ında  ğrenilir.

** st D ğ m ATM U  Sistem Adresi :**  st D ğ m n ATM U  Sistem Adresi.  st D ğ m ATM U  Sistem Adresi, yeterli d ğ msel hiyerar i listesi olan Hellolar kom u d ğ mden alındı ında  ğrenilir.

**T retilen Ba  Toplanma G stergesi :** Bir ba  i in ba  toplanma g stergesinin algoritmik olarak t retilen de eri.

**Konfig re Edilmi  Ba  Toplanma G stergesi :** Bir ba  i in d ğ m n konfig re edilmi  toplanma g stergesi.

**Uzak Ba  Toplanma g stergesi :** Ba ın di er ucunda bulunan kom udan Hellolarla alınan ba  toplanma g stergesi.

#### **4.1.1.2 Hello Durumları**

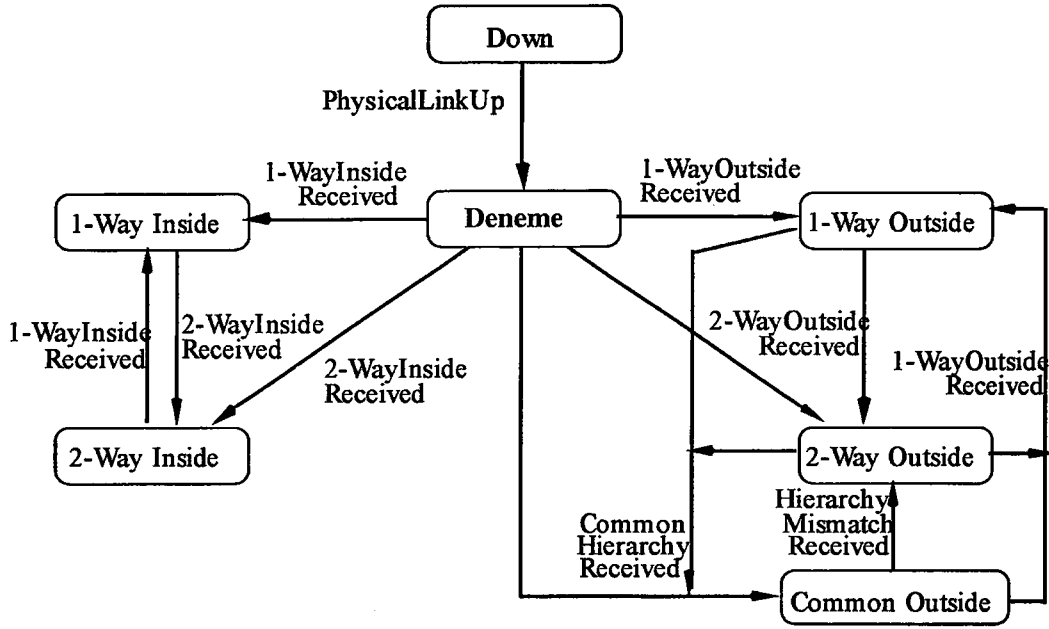
Bu b l mde Hello Sonlu Durum Makinasının (FSM) ula abilece i durumlar anlatılmı tır.  ekil 4.1'de olası durum de i iklikleri g sterilmi tir. Yayların  zerindeki etiketler, durum de i imine sebep olan olayları belirtirler.

 ekil 4.1 de belirtilen durum ge i lerine ek olarak  ekilde g sterilmeyen bazı ek durumlarda mevcuttur. Bunlar:

LinkDown olayı, her zaman Down durumuna ge i i mecbur kılar,

Hareketsizlik Zamanlayıcısı olayı, her zaman Deneme Durumuna ge i i mecbur kılar,

HelloMismatchReceived olayı, her zaman Deneme Durumuna ge i i mecbur kılar.



Şekil 4.1 Hello Durum Değişiklikleri

**Down Durumu :** Hello FSM'nin başlangıç durumu. Ayrıca, alt-seviye protokolleri bağlantının kullanılamaz olduğunu belirttiği durumlarda da bu duruma varılır. Bu tür bir bağ üzerinden PNNI yönlendirme paketleri gönderilmez veya alınmaz.

**Deneme :** Bu durum, komşudan son zamanlarda ya hiçbir Hellonun alınmadığını ya da yanlış (uygunsuz) bilgili bir Hellonun alındığını belirtir. Bu durumda, komşuya periodik olarak, Hello Aralığı uzunluğunda periodlarla Hellolar gönderilerek bağlantı kurmak için denemeler yapılır.

**1-Way Inside :** Bu durumda, Hellolar komşudan henüz alınmıştır ve her iki düğümün de aynı grubun üyesi olduğu anlaşılmıştır, fakat komşunun Hellolarındaki uzak düğüm kimliği ve uzak kapı kimliği sıfıra ayarlanmıştır.

**2-Way Inside :** Bu durumda, her iki düğümün de aynı peer grubun üyesi olduğunu ve doğru uzak düğüm kimliği ve uzak kapı kimliği alanları içerdiğini belirten Hellolar komşudan henüz alınmıştır. Bu duruma ulaşıldığında, iki düğüm arasında bu bağ üzerinden iki-yönlü iletişim elde edilmiş demektir. Veritabanı Özet Paketleri, PTSE Talep paketleri, PTSP'ler ve PTSE onay paketleri sadece 2-Way Inside durumunda yeralan bağlar üzerinden gönderilebilir.

**1-Way Outside :** Bu durum, Helloların komşudan henüz alındığını ve komşu düğümün başka bir peer gruba üye olduğunu belirtir, ama komşunun Hellolarındaki uzak düğüm kimliği ve uzak kapı kimliği sıfıra ayarlanır. Bu durumda ve 2-Way Outside durumunda düğüm, hem bu düğümü hem de komşu düğümü içeren bir ortak peer grup arar.

**2-Way Outside :** Bu durumda, Hellolar komşudan henüz alınmıştır ki bu da komşunun farklı bir peer gruba ait olduğunu, doğru uzak düğüm kimliğini ve uzak kapı kimlik alanlarını içerdiğini belirtir. Fakat düğümsel hiyerarşi listesi herhangi bir ortak peer grubu kapsamaz. Bu durumda ve 1-Way Outside durumunda düğüm, hem bu düğümü hem de komşu düğümü içeren bir ortak peer grup arar.

**Common Outside :** Bu duruma ulaşıldığında, iki düğüm arasında iki-yönlü tam iletişimin elde edilmesinin yanında, ortak bir yönlendirme hiyerarşi seviyesi de bulunulmuş demektir.

#### **4.1.1.3 Hello Durum Değişimlerine Yol Açan Olaylar**

Durum değişiklikleri, Hello protokolünün işletilmesi ile ilgili birçok olası olaylarla ortaya çıkabilir. Bu olaylar şekil 4.1’de etiketli yaylar olarak gösterilmiştir.

**LinkUp :** Alt katman protokolleri bağın işlevsel duruma geçtiğini belirtmişlerdir.

**1-WayInsideReceived :** Komşudan, peer grup kimliği bu düğümün peer grup kimliği ile uyuşan, bir hello alınmıştır. Bu durumda Uzak Düğüm kimliği ile Uzak Kapı kimliği sıfıra ayarlanır. Bunlara ek olarak, bu hello veri yapısındaki Sürüm, Uzak Düğüm Kimliği, Uzak Kapı Kimliği ve Uzak Peer Grup Kimliği alanları sıfıra eşit değilse, Hellodan alınan Peer Grup Kimliği, Kapı Kimliği, Kaynağın Düğüm Kimliği ve Protokol Sürümü ile uyuşmalıdır.

**2-WayInsideReceived :** Komşudan bir Hello alınmıştır, bu Hellodaki uzak düğüm kimliği ve uzak kapı kimliği, bu düğümün düğüm kimliğini ve kapı kimliğini doğru olarak tanıtır ve peer grup kimliği bu grubun düğüm peer grup kimliği ile uyuşur. Bunlara ek olarak eğer hello veri yapısındaki Sürüm, Uzak Düğüm Kimliği, Uzak Kapı Kimliği ve Uzak Peer Grup Kimlik alanları sıfıra eşit değilse, Hellodan alınan Peer Grup Kimliği, Kapı Kimliği, Kaynağın Düğüm Kimliği ve Protokol Sürümü ile uyuşmalıdır.

1-WayOutsideReceived : Komşudan bir Hello alınmıştır, bu Hellonun içinde bu düğümün kendi peer grup kimliği ile uyuşmayan bir peer grup kimliği bulunur ve uzak düğüm kimliği ile uzak kapı kimliği sıfıra ayarlanmıştır. Bunlara ek olarak, Hello veri yapısındaki Sürüm, Uzak Düğüm Kimliği, Uzak Kapı Kimliği ve Uzak Peer Grup Kimliği alanları sıfıra eşit değilse Hellodan alınan Peer grup kimliği, Kapı kimliği, Kaynağın Düğüm Kimliği ve protokol Sürümü ile uyuşmalıdır.

2-WayOutsideReceived : Komşudan aşağıdakileri içeren bir Hello alınmıştır:

- (i) bu bağ için bu düğümün düğüm kimliğini ve Kapı kimliğini doğru olarak yansıtan uzak düğüm kimliği ve uzak kapı kimlik alanları,
- (ii) bu düğümün kendi peer grup kimliği ile uyuşmayan peer grup kimliği,
- (iii) Ortak peer grup kimliği, Üst düğümün Düğüm kimliği ve hello veri yapısındaki Üst düğüm ATM Uç Sistem Adresleri sıfıra ayarlanmıştır,
- (iv) Aşağıdakilerden biri mevcuttur(doğrudur):

A – Ya düğümsel hiyerarşi listesi, ya toplanma göstergesi ya da ULIA yoktur

B – düğümsel hiyerarşi listesi vardır ama bir ortak peer grup içermez.

Ayrıca, eğer Sürüm, Uzak düğüm kimliği, Uzak Kapı kimliği ve hello veri yapısındaki Uzak Peer Grup kimlik alanları sıfıra eşit değilse sırasıyla Protokol sürümü, Kaynağın Düğüm Kimliği, Kapı Kimliği ve alınan Hellonun Peer grup kimliği ile uyuşmalıdır.

CommonHierarchyReceived : Komşudan aşağıdakileri içeren bir Hello alınmıştır:

- (i) bu bağ için bu düğümün düğüm kimliğini ve Kapı kimliğini doğru olarak yansıtan uzak düğüm kimliği ve uzak kapı kimlik alanları,
- (ii) bu düğümün kendi peer grup kimliği ile uyuşmayan peer grup kimliği,
- (iii) bir ULIA
- (iv) bir Toplanma göstergesi

(v) aşağıdakilerden birini içeren hiyerarşi listesi:

- alınan Döğümsel Hiyerarşi Sıra numarasına uyan bir sıra numarası ve hello veri yapısına önceden kurulmuş bir Ortak peer Grup kimliği, veya
- bir ortak peer grup ve hello veri yapısındaki ortak peer grup kimliği, Üst düğümün Döğüm kimliği ve Üst düğüm ATM Uç Sistem Adresleri sıfıra ayarlanmıştır veya bu alanlar sırasıyla alınan döğümsel hiyerarşi listesindeki en düşük peer grup ile ve bu ortak peer grup kimliği ile ilgili ATM Uç Sistem Adresleri ve Döğüm kimliği ile uyuşmalıdır.

Bunlara ek olarak, eğer hello veri yapısındaki Uzak Peer Grup kimlik alanları, Uzak Kapı Kimliği, Uzak düğüm kimliği ve Sürüm sıfıra eşit değilse sırasıyla Protokol Sürümü, Kaynağın Döğüm Kimliği, Kapı Kimliği ve alınan Hellodaki Peer Grup Kimliği ile uyuşmalıdır.

HelloMismatchReceived : İçindeki uzak düğüm kimliği ve/veya uzak kapı kimliği düğümün kendi düğüm kimliği veya alan bağı için kapı kimliğinden farklı olan ve her ikisi de sıfıra ayarlı bir Hello alınmıştır.

HierarchyMismatchReceived : Komşudan bir Hello alınmıştır ve bu Hello:

- (i) bu düğümün, düğüm kimliğini ve bu bağı için kapı kimliğini doğru olarak yansıtan bir uzak düğüm kimliği ve uzak kapı kimlik alanlarını içerir,
- (ii) bu düğümün kendi peer grup kimliği ile uyuşmayan bir peer grup kimliği içerir,
- (iii) HelloMismatchReceived kriterini karşılamaz,
- (iv) Aşağıdaki durumlardan biri olur :
  - Bir döğümsel hiyerarşi listesi yoktur
  - Bir ULIA bulunmamıştır

- Bir toplanma göstergesi bulunmamıştır
- Yeni bir düğümsel hiyerarşi liste sıra numarası alınmıştır ya da,
  - peer grup seviyelerinin sırası değer olarak birbirini takip etmemektedir,
  - ortak bir peer grup bulunmamıştır,
  - bulunan en düşük ortak peer grup kimliği, hello veri yapısındaki Ortak Peer Grup Kimliğinden farklıdır,
  - Ortak Peer Grup Kimliği içeren bilgi üçlüsünde, bir yüksek seviye düğüm kimliği veya hello veri yapısındaki Üst düğümün Düğüm Kimliği veya ATM Uç Sistem Adresi ile uyuşmayan yüksek seviye ATM Uç Sistem Adresi bulunur.

**HelloTimerExpired** : Hello Zamanlayıcısı sona erdi.

**InactivityTimerExpired** : Eylemsizlik Zamanlayıcısı sona erdi. Yani, yakın zamanda komşudan bir hello alınmadı.

**LinkDown** : Alt katman protokolleri, bu bağı kullanılamayacağını belirtirler.

#### **4.1.2 Helloların Gönderilmesi**

Down durumu dışındaki her durumda, Hellolar her Hello Aralık süresinin dolumunda periyodik olarak gönderilirler. Buna ek olarak, olayla-tetiklenen Hellolar aşağıdaki durumlarda gönderilir:

1. Aşağıdaki durum değişimleri hariç, her durum değişiminde:

- 1-Way Inside durumundan 2- Way Inside durumuna
- 1- Way Outside durumundan 2- Way Outside durumuna
- 1- Way Outside durumundan Common Outside durumuna
- 2- Way Outside durumundan Common Outside durumuna ve

- Common Outside durumundan 2- Way Outside durumuna

2. Dış bağ için ULIA'da her önemli değişiklik ortaya çıktığında bu dış bağda.
3. Düğümün hiyerarşi listesinde her önemli değişiklik ortaya çıktığında bir dış bağda.
4. Dış bağ için bağ toplanma göstergesinde her önemli değişiklik ortaya çıktığında bu dış bağda.

Helloların ardışık gönderilmeleri arasında geçen süre minimum Hello Aralığından kısa olamaz. Bekleme zamanlayıcısının kullanılması düğümün kabul edilemeyecek kadar yüksek oranlarda Hellolar göndermesini engeller. Eğer bekleme zamanlayıcısı yüzünden birkaç olayla-tetiklenen hello erteleniyorsa, bekleme zamanlayıcısı sona erdiğinde tüm IG'ler için en yeni bilgileri içeren sadece bir hello gönderilir. Olayla-tetiklenen helloların her gönderilişinde Hello zamanlayıcısı başa döner, böylece diğer hello Hello Aralığından sonra görevlendirilir.

Hello durumu Deneme iken, Helloların uzak düğüm kimlikleri ve uzak kapı kimlik alanları hello veri yapısında saklandığı şekilde sıfıra ayarlanmalıdır. Hello durumu 1-Way Inside, 2-Way Inside, 1-Way Outside, 2-Way Outside veya Common Outside iken Hellolar, uzak düğüm kimliği ve uzak kapı kimliği olarak, komşu düğümün düğüm kimliği ve kapı kimliğini içermelidir. Bunlar aynı bağ üzerinden alınmış Hellolardan öğrenilir ve hello veri yapısı içinde saklanır.

Düğümün tüm üst seviye düğüm kimliklerini, peer grup kimliklerini ve Üst Seviye Bağlama bilgisindeki PGL'den aldığı şekilde LGN ATM Uç Sistem adresini anlatan düğümsel hiyerarşi listesi, durum 1-Way Outside, 2-Way Outside ve Common Outside iken Hellolara dahil edilir. Eğer birden fazla düğüm bu peer grubun liderliğini talep ederse yerel olarak peer grup lideri seçilen bilgiyi ilan eder.

Bilinen üst seviyelerin sayı veya içeriğinde meydana gelen her değişiklikte düğümsel hiyerarşi listesinde belirtildiği gibi, düğümsel hiyerarşi listesinin sıra numarası arttırılmalı ve bir olayla-tetiklenen Hello gönderilmelidir. Bunun dışında, düğüm kimliğinde, peer grup kimliğinde veya en düşük seviyedeki ATM adresinde meydana gelen her değişiklikte düğümsel hiyerarşi listesinin sıra numarası arttırılmalı ve bir

olayla-tetiklenen Hello gönderilmelidir. Komşuya düğümsel hiyerarşi listesinin ilk örneği gönderilirken sıra numarasının değeri sıfırdan büyük olmalıdır.

Düğümsel hiyerarşi listesi içeren bir Hellonun her gönderilişinde, listede o an bilinen tüm yüksek seviyeler bulunmalıdır. Eğer hiçbir üst seviye hiyerarşisi bilinmiyorsa boş bir hiyerarşi listesi eklenmelidir. Hiyerarşi listesi üretilirken, düğümsel hiyerarşi listesine dahil olan seviye sayısı, her ardışık Hello ile artabilir.

#### **4.1.2.1 ULIA'ların Oluşturulması**

Belli Hello durumları için, bir sınır düğümü, komşu sınır düğümlerine dış bağlar üzerinden gönderdiği Hello paketlerine ULIA IG eklemelidir. ULIA herhangi bir bağ durum bilgisi belirtmez, daha çok diğer IG'leri birleştirmek için kullanılır. Bu mekanizmanın sağladığı esneklik sayesinde bir sınır düğümü, komşusu tarafından ilan edilen tam bağ durum bilgisini (IG'ler), komşusunun kişisel IG'leri anlamasına veya yorumlamasına gerek duymadan, dikte edebilir. Bu yolla, ağ bölümler halinde güncellenebilir (örneğin, yeni ve bilinmeyen IG'lerde yeni güvenlik tedbirleri ve politikalar sağlamak için), ve hiyerarşiden ilan edilen doğru bağ durum bilgisine sahip olabilir.

ULIA içeriğindeki önemli değişiklikler, ULIA sıra numarasındaki değişikliklerle belirtilir. Başlatıcı, ULIA içeriğindeki değişiklik önemli olduğunda sadece ULIA sıra numarasını artırır.

Sınır düğümleri Hello paketleriyle alınan ULIA'lar içindeki kişisel IG'leri yorumlamadığından ULIA sıra numarasındaki herhangi bir değişiklik, önemli bir değişikliği belirtir ve bu yüzden ilgili üst bağ PTSE'ye bir güncelleme tetiklemelidir. Tetiklenen bu güncellemeye cevap olan üst bağ PTSE'nin yeni örneğinin baştan üretilme işlemi, PTSE bekleme zamanlayıcısına tabidir.

Bir sınır düğümü, ULIA içinde bulunan IG'lerin bir veya daha fazlasında önemli değişikliklerin meydana geldiği durumlar dışında, Hellolarla gönderdiği ULIA için yeni sıra numaraları üretmemelidir. Son ULIA'nın Hello paketiyle gönderilmesinden bu yana önemli bir değişiklik meydana gelmemişse, önceki sıra numaralı ULIA (muhtemelen değiştirilmiş) Hellolarla gönderilmeye devam etmelidir. Paketlenmiş IG'lerdeki önemsiz değişiklikler, hiyerarşik yayılmanın masraflarına katlanacak

kadar önemli değildir ve bu yüzden sıra numarasında bir değişiklik yapılmaz. Önemli bir değişikliğe tepki olarak yeni bir ULIA'nın her oluşuşunda sıra numarası arttırılmalıdır.

Önceden bir ULIA içinde paketlenmiş ve bir Hello paketi ile gönderilmiş herhangi bir IG için önemli bir değişiklik olayı yeni bir ULIA'nın oluşmasını ve yeni Hello paketinin ilgili dış komşuya gönderilmesini tetikler. Yeni Hello paketi gönderildiğinde yeni ULIA ve sıra numarasını içerir. Bir veya daha fazla kez paketlenmiş ULIA'daki önemli değişikliğe tepki olarak ortaya çıkan yeni ULIA oluşturulurken sınır düğüm en yeni ve en doğru bağ durum bilgisini tüm paketlenmiş IG'lere, bunlarda önemli bir değişiklik olup olmamasına bakmaksızın sokar. Bu yolla, herhangi bir IG'deki önemli bir değişiklik tüm paketlenmiş IG'ler için en yeni bilgilerin aynı ULIA'da ilan edilmesine yol açar.

#### 4.2 Veritabanı Zamanuyumlaması

Bir düğüm, komşu bir peer düğümünün varlığını (aynı peer grup içinde bulunan) ilk öğrendiğinde, komşu peer'ların topoloji veri tabanlarının zamanuyumlamasını yapmak için bir veritabanı değişim işlemini başlatır. Veritabanı değişim işlemi, bir düğümün topoloji veritabanındaki tüm PTSE'leri tanıtan bilgileri içeren Veritabanı Özet Paketleri dizisinin karşılıklı değişimidir. Veritabanı Özet Paketleri kilit-adım mekanizması kullanılarak değiştirilir, bunun sayesinde bir taraf Veritabanı Özet Paketi gönderirken diğer taraf kendi Veritabanı Özet Paketi ile cevap verir (alınan paketi dahili olarak onaylayarak). Herhangi bir anda iki komşu arasında en fazla bir geçerli pakete izin verilir.

Bir düğüm, komşu peer'dan bir Veritabanı Özet Paketi aldığı anda topoloji veritabanını, pakette anlatılan tüm PTSE'leri bulmak için inceler. Eğer topoloji veritabanında PTSE bulunmazsa veya komşu peer'da daha yeni bir PTSE sürümü varsa düğüm komşudan bu PTSE'yi istemelidir veya isteğe bağlı olarak, veritabanı özetinde en yeni PTSE sürümü olduğu anlaşılan başka bir komşu peer'dan da PTSE istenebilir.

En düşük seviye komşu peer'lar için birden fazla paralel fiziksel bağ bulunabilir ve/veya bunların arasında VPC'ler olabilir. Her fiziksel bağlantı ve/veya iki komşu

peer arasındaki VPCler için farklı bir durum makinesi kullanacaklardır. Ama, veritabanı zamanuyumlaması ve taşkın amaçlı olarak, komşu peer'lar arasında sadece bir görüşme yapılacaktır. Bu görüşme, komşu peer durum makinesi tarafından ve komşu peer'a veritabanı zamanuyumlaması ve taşkın sağlamak için ihtiyaç duyulan komşu peer veri yapısı tarafından tarif edilir. Bir bağda, 2-WayInside Hello durumuna her ulaşıldığında, ilgili komşu peer durum makinesinde AddPort olayı tetiklenir. Benzer şekilde, bir bağ 2-WayInside durumundan düştüğünde ilgili komşu peer durum makinesinde DropPort olayı tetiklenir. AddPort olayı tetiklendiğinde, iki komşu peer arasında bağ oluştuktan sonra, veritabanı değişim işlemi başlar. İki komşu peer arasındaki son bağ için DropPort olayı ortaya çıktığında, komşu peer durum makinesi, komşu peer için tüm durum bilgilerinin silinmesine yol açan DropPort olayını dahili olarak üretir.

PTSPlar, PTSE Alındı paketleri, Veritabanı Özet Paketleri veya PTSE Talep paketleri gönderildiğinde, 2-WayInside Hello durumunda olan komşu peer'lar arasındaki bağlardan herhangi biri kullanılabilir. PNNI yönlendirme bilgilerinin dağıtım ve bakımına bir zarar vermeksizin, ardışık paketler farklı bağlara gönderilebilir. En düşük seviye komşu peer'lar arasındaki bağlar, sadece komşudan gelen peer durum makinesi Dolu durumunda olduğunda tanıtılabilir. Komşu en düşük seviye peer'ların fiziksel bağlar ve VPC'lerle bağlandığı durumlar için, Dolu durumunun içine veya dışına doğru yapılan değişiklikler, bu düğümün PTSE'lerinden birinin veya birkaçının yeni örneklerinin üretilmesine yol açar.

Komşu peer mantıksal grup düğümleri arasında, sadece SVCC-tabanlı RCC'ler PNNI yönlendirme paketlerinin değişimi için kullanılır. Benzer şekilde, en düşük seviye komşu peer'ların durumunda da, komşu peer durum makinesi RCC'nin Hello durum makinesi ile birleştirilir. LGN'ler arasındaki yatay bağların Hello durumları, komşu peer durumunu etkilemez. RCC'nin Hello durumu 2-WayInside durumuna ulaştığında komşu peer durum makinesinde AddPort olayı tetiklenir ve veritabanı değişimi başlar. RCC'nin Hello durumu 2-WayInside durumundan düşerse komşu peer durum makinesinde DropPort olayı tetiklenir ve bu olayın sonucunda Dolu durumundan NPDown durumuna geçer.

Komşuların SVCC-tabanlı RCC ile iletişim kurduğu durumlarda komşu peer durum makinesi yatay bağ PTSE'lerin oluşumunu direkt olarak etkilemez. Bunun yerine,

yatay bađ PTSE'lerin oluřumunu dolaylı yoldan Yatay Bađ Hello protokolü ile etkiler. Buna ek olarak, bir bađ için bir PTSE ilk olarak oluřturulurken ilgili Mantıksal Dügüm (LGN) Hello makinesi 2-WayInside durumunda ve peer veri yapısı da Dolu durumunda olmalıdır.

#### 4.2.1 Komřu Peer Veri Yapısı

Her düğümün tüm komřu peer düğümleri için, bu düğümler arasındaki bađ sayısına bakılmaksızın, birer komřu peer veri yapısı vardır. NPDown durumu dışındaki tüm durumlarda, komřu peer görüşmelerine "komřuluk" adı verilir. Komřu peer veri yapısı içinde, iki komřu peer arasında komřuluk oluřturmak için gerekli olan tüm bilgiler veya daha önceden oluřturulmuř komřuluk hakkındaki tüm bilgiler bulunur. Farklı peer gruplara ait komřu düğümler komřuluk oluřturmayacaklardır.

Durum : Komřu peer FSM'nin durumu.

Uzak Dügüm Kimliđi : Komřu peer düğümünü tanımlamak için kullanılan düğüm Kimliđi.

Kapı Kimliđi Listesi : Kapı Kimliđi Listesi, sadece fiziksel bađlar ve/veya VPC'ler ile birleřtirilmiř en düşük seviye komřu peer'ların bulunduđu durumda kullanılır. Kapı Kimliđi Listesi, komřu peer'a giden 2-WayInside durumunda olan bađların listesidir. PTSP'ler, PTSE alındı paketleri, veritabanı özet paketleri veya PTSE talep paketleri komřu peer'a gönderildiđinde veya tekrar gönderildiđinde, bu listede belirtilen bađlardan herhangi biri kullanılabilir.

Efendi / Köle : İki komřu peer veri tabanlarını deđiřtirirken bir efendi / köle iliřkisi oluřtururlar. Bu iliřki sadece bařlangıç topoloji veritabanı deđiřimi ile ilgilidir. Efendi, ilk Veritabanı Özet Paketini gönderip ilk DS Sıra Numarasını seđer. Veritabanı Özet paketlerinin yeniden-gönderimleri Efendi tarafından kontrol edilir. Köle sadece Efendinin Veritabanı Özet Paketlerine cevap verebilir. Efendi / Köle iliřkisi PazarlıkEdiliyor durumunda belirlenir.

DS Sıra Numarası : Kiřisel veritabanı özet paketlerini tanımlayan iřaretsiz 32-bitlik bir rakamdır. PazarlıkEdiliyor durumuna ilk girildiđinde DS Sıra Numarası, komřu peer tarafından daha önce kullanılmamıř bir rakama ayarlanmalıdır. DS sıra

numarası Efendi tarafından her Veritabanı Özet Paketinin gönderilmesiyle artırılır. Kölenin DS sıra numarası, Efendiden alınan son paketi belirtir.

Peer Yeniden Gönderme Listesi : Taşkın yöntemiyle dağıtılan, fakat komşu peer tarafından onaylanmayan PTSE'lerin listesidir. Bunlar onaylanana kadar veya komşu peer durum makinesi tarafından boşaltılana kadar periyodik olarak gönderilmeye devam edilirler.

Bu listedeki her girişe ait bir PTSE Yeniden Gönderme Zamanlayıcısı vardır. Bu, PTSE Yeniden Gönderme Aralığı sonunda tetiklenen bir PTSE Yeniden Gönderme Zamanlayıcısıdır. Bu PTSE'ye uyan bir onay alındığında, zamanlayıcı durdurulur.

PTSEYenidenGöndermeAralığı : Onaylanmayan her PTSE, her PTSEYeniden GöndermeAralığı dolduğunda yeniden gönderilir.

Peer Gecikmiş Alındı Listesi : PTSE'ler için komşu peer'a gönderilecek gecikmiş onay paketleri listesidir. Her PeerGecikmişAlındıAralık saniyesinde, alındı paketleri, Peer Gecikmiş Alındı Listesindeki tüm girişleri için PTSE tanıtıcı bilgi içeren komşu peer'a gönderilir ve liste temizlenir.

PeerGecikmişAlındıAralığı : Peer Gecikmiş Alındı listesinin ardışık kontrolleri arasında geçen zaman aralığıdır.

Peer Gecikmiş Alındı Zamanlayıcısı : Bu zamanlayıcı tükendiğinde Peer Gecikmiş Alındı Listesinde onaylanmayan tüm PTSE'ler bir onay paketinde toplanıp komşu peer'a gönderilir.

PTSE Talep Listesi : Komşu iki peer'ın topoloji veri tabanlarının zamanuyumlamasını yapmak için gereken PTSE'lerin listesidir. Bu liste, Veritabanı Özet Paketleri alındıkça oluşturulur. PTSE Talep paketleri, bu komşu peer'dan veya isteğe bağlı olarak, eksik PTSE'ye sahip olduğu bilinen herhangi başka bir komşu peer'dan gelen bu listedeki tüm PTSE'leri talep etmek için kullanılır. Uygun PTSE'ler alındıkça liste boşaltılır.

DSRxmtAralığı : Bir düğümün önceki Veritabanı özet paketini tekrar göndermeden önce beklediği süredir.

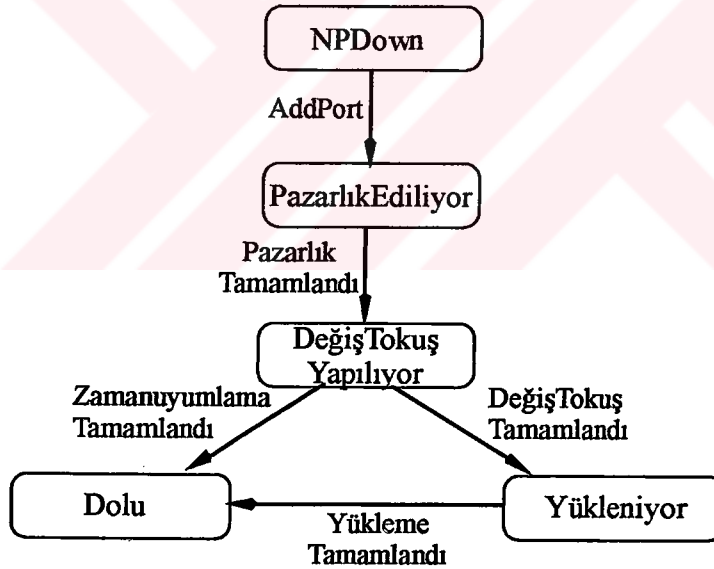
DS Rxmt Zamanlayıcısı : DSRxmtAralığı sonunda tetikleyen bir zamanlayıcıdır. Düğüm doğru bir Veritabanı Özet Paketi aldığı anda zamanlayıcı durdurulur.

TalepRxmtAralığı : Düğümün, henüz alınmamış son PTSE Talep Paketinin PTSE'lerini talep eden, yeni bir PTSE Talep Paketini göndermeden önce beklediği süredir. (Saniye ile belirtilir).

Talep Rxmt Zamanlayıcısı : TalepRxmtAralığı sonunda tetikleyen bir zamanlayıcısıdır. Son PTSE Talep Paketinde talep edilen tüm PTSE'ler alındığında zamanlayıcı durdurulur.

#### 4.2.2 Komşu Peer Durumları

Komşu peer durum makinesi, komşu peer'da devam eden taşkını ve veritabanı zamanuyumlamasının durumunu anlatır. Şekil 4.2'de olası durum değişiklikleri gösterilmiştir. Şekildeki yaylar, durum değişikliğine sebep olan olaylarla etiketlenmiştir.



Şekil 4.2 Komşu Peer Durum Değişiklikleri (Veritabanı Zamanuyumlaması)

Şekilde gösterilen durum geçişlerine ek olarak:

DSMismatch olayı, PazarlıkEdiliyor durumuna geçişi mecbur kılar,

BadPTSERequest olayı, PazarlıkEdiliyor durumuna geçişi teşvik eder,

DropPort olayı hiçbir durum değişikliğine neden olmaz,

DropPortSon olayı, NP Down durumuna geçişi teşvik eder.

NP Down : Komşu peer FSM'nin ilk durumu. Bu durum, komşu peer'a faal bir bağ bulunmadığını gösterir (örn. Hello durumunda 2-WayInside). Bu durumda, komşu peer ile ilgili bir komşuluk yoktur.

PazarlıkEdiliyor : İki komşu peer arasında komşuluk yaratmanın ilk adımıdır. Bu adımın amacı hangi düğümün efendi, hangi düğümün köle olduğuna karar vermek ve ilk sıra numarasını belirlemektir.

DeğişTokuşYapılıyor : Bu durumda düğüm, komşu peer'a Veritabanı Özet Paketleri göndererek topoloji veritabanını tarif eder. Veritabanı Özet Paketlerinin işletilmesinin sonucu olarak ihtiyaç duyulan PTSE'ler talep edilebilir.

Yükleniyor : Bu durumda, Veritabanı Özet Paketleri komşu peer ile karşılıklı değiştirilmiş, ihtiyaç duyulan PTSE'ler talep edilmiştir ve en azından bir tane PTSE henüz alınmamıştır.

Dolu : Bu durumda, düğüm komşu peer'da varolduğu bilinen tüm PTSE'leri almıştır.

#### **4.2.3 Komşu Peer Durum Değişiklerine Yol Açan Olaylar**

Durum değişikliğine birçok olay sebep olabilir. Bu olaylar, iki komşu peer arasındaki veritabanı zamanuyumlaması ile ilgili prosedürler ya da ilgili bağlar için Hello durum makinesinin hareketleriyle tetiklenebilir. Şekil 4.2'de etiketli yaylarla gösterilen olaylar aşağıdaki şekilde tanımlanırlar:

AddPort : Komşu bir peer'a olan bağın Hello durum makinesi 2-WayInside durumuna ulaşmıştır.

PazarlıkTamamlandı : Efendi / Köle ilişkisi görüşüldü ve ilk veritabanı özet paketi sıra numarası üzerinde anlaşıldı.

DeğişTokuşTamamlandı : Komşu peer'in son Veritabanı Özet Paketi alındı, bu düğümün son Veritabanı Özet Paketi gönderildi (PTSE talep listesi boş değil). Bu andan sonra düğüm hangi PTSE'nin talep edileceğini bilir.

**ZamanUyumlamaTamamlandı :** Komşu peer'in son Veritabanı Özet Paketi alındı, bu düğümün son Veritabanı Özet Paketi gönderildi. ( PTSE talep listesi boş değil).

**YüklemeTamamlandı :** PTSE Talep Listesindeki en son PTSE alındı.

**DSMismatch :** Bir Veritabanı Özet Paketi alındığında, bu pakette:

- a) beklenmeyen bir veritabanı özet paketi sıra numarası varsa,
- b) beklenmeyen Başlatma bit ayarı var, veya
- c) beklenmeyen Efendi bit ayarı varsa,

bu durumlardan herhangi biri, veritabanı zamanuyumlamasında bir hata oluştuğunu gösterir.

**BadPTSERequest :** Veritabanında bulunmayan bir PTSE için, bir PTSE Talebi alındığında veya alınan PTSE, PTSE Talep Listesindekinden daha eski olduğu durumda, veritabanı zamanuyumlamasında bir hata oluştuğu gözlemlenir.

**DropPort :** Komşu peer'a olan bağın Hello durum makinesi 2-WayInside durumundan çıktı.

**DropPortLast :** DropPort olayı çalıştırılırken bu komşuya tüm kapıların düştüğüne karar verildi.

#### **4.2.4 Veritabanı Özet Paketlerinin Gönderilmesi**

Bu bölümde Veritabanı Özet paketlerinin komşu peer gruba nasıl gönderildiği anlatılmıştır.

Herhangi bir anda sadece bir tane geçerli Veritabanı Özet paketine izin verilir.

Veritabanı Özet paketlerinin gönderilmesi komşu peer'ın durumuna bağlıdır.

PazarlıkEdiliyor durumunda düğüm boş Veritabanı Özet paketini, Başlangıç, DahaFazla ve Efendi bit ayarları ile gönderir. Bu tür Veritabanı Özet paketleri gönderilirken, Ds Rxmt Zamanlayıcısı tekrar başlatılmalıdır. Bu paketler her DSRxmt Zamanlayıcısı tetiklediğinde, her DSRxmt Aralığı sonunda tekrar gönderilirler.

DeğişTokuşYapılıyor durumunda Veritabanı Özet Paketleri, düğüme ait veritabanında bulunan topoloji durum bilgisinin özetlerini içerir. Mantıksal Grup Düğümleri durumunda, Mantıksal Grup Düğümleri seviyesinde veya daha yüksek seviyelerde oluşturulan veya bu seviyelerde alınan bu topoloji veritabanı bölümleri, veritabanı özetine dahil edilir (düşük seviye PTSE'ler, bir anahtarlama sisteminin topoloji veritabanına bir veya daha fazla oluşumu için bir düşük seviye düğümü olarak ait olabilir, ama mantıksal grup düğümünün topoloji veritabanına ait olmazlar). Bu tür bir PTSE'nin PTSP ve PTSE başlık bilgisi, düğümün Veritabanı Özet paketlerinin birinde listelenir. DeğişTokuşYapılıyor durumuna girildikten sonra adına yeni örnekler alınan PTSE'ler, herhangi bir Veritabanı Özet Paketine dahil edilmek zorunda değildirler, çünkü normal taşkın prosedürlerine göre ele alınacaklardır. Her PTSE'nin komşu peer'a gönderilen Veritabanı Özet paketler dizisine en fazla bir kere dahil edilmesi tavsiye edilse de şart değildir.

DeğişTokuşYapılıyor durumunda bir Veritabanı Özet paketinin ne zaman gönderileceği düğümün efendi veya köle oluşuna bağlıdır. Yeni bir Veritabanı Özet paketi gönderileceği zaman, paketin veritabanı özet paketi sıra numarası ayarlanır ve düğümün topoloji veritabanından gelen, bir grup yeni PTSE tanımlanır. Kullanılan kurallar düğümün efendi veya köle oluşuna bağlı olarak değişiklik gösterir:

Efendi : Veritabanı Özet paketlerinin gönderilme durumları :

- (i) Köle, önceki Veritabanı Özet paketini aldığını, DS sıra numarasını yansıtarak bildirir,
- (ii) DSRxmt Aralık saniyeleri boyunca bir alındı olmaması durumunda önceki Veritabanı Özet Paketi yeniden gönderilir. Bir Veritabanı Özet Paketinin her gönderilişinde DSRxmt Zamanlayıcısı yeniden başlatılmalıdır. Düğüm, Veritabanı Özet paketlerinin tamamını zaten göndermişse veya pakette köleye gönderilecek veritabanı özet paketlerinin son kısımları varsa, DahaFazla bit ayarı sıfıra eşitlenmelidir.

Köle : Veritabanı Özet paketleri, sadece efendiden alınan Veritabanı Özet paketlerine cevap olarak gönderilirler. Efendiden alınan paket yeniyse, yeni bir Veritabanı Özet paketi gönderilir; aksi taktirde önceki Veritabanı Özet paketi yeniden gönderilir. Eğer düğüm, Veritabanı Özet paketlerinin tamamını göndermişse, (örneğin,

Veritabanı Özet paketinin içeriği boşsa) DahaFazla bit ayarı sıfıra ayarlanmalıdır. Yükleniyor ve Dolu durumlarında köle son Veritabanı Özet paketini, efendiden alınan kopya Veritabanı Özet paketlerine cevap olarak tekrar göndermelidir. Yükleniyor ve Dolu durumlarında kölenin gönderdiği son paket boş olmalıdır, Başlangıç, DahaFazla ve Efendi bit ayarı sıfıra ayarlı olmalıdır ve o anki peer veri yapısındaki DS sıra numarasının aynısına sahip olmalıdır.

#### 4.2.5 Veritabanı Özet Paketlerinin Alınması

Bu bölümde, alınan Veritabanı Özet paketinin işlenmesi detaylı olarak anlatılmaktadır. Her bir Veritabanı Özet paketinin bir sıra numarası vardır. Veritabanı Özet paketinin ileriki işlemleri komşu peer veri yapısının Uzak Düğüm Kimliği ile ilgili durumuna bağlıdır. Eğer komşu peer durumu NPDown ise paket dikkate alınmamalıdır.

PazarlıkEdiliyor : Eğer alınan paket aşağıdaki durumlardan birine uyarsa, komşu peer durum makinesi PazarlıkTamamlandı durumunda işletilmelidir (durumun DeğişTokuşYapılıyor durumuna geçmesine sebep olur) ve paket sırada bir sonraki olarak kabul edilip işlem devam ettirilmelidir. Aksi halde, paket yok sayılmalıdır.

- Başlangıç, DahaFazla ve Efendi bitleri sıfırdır, paketin içeriği boştur ve komşu peer'in Düğüm Kimliği bu düğümün kendi düğüm kimliğinden büyüktür. Bu durumda bu düğüm şimdi bir Köle'dir. PazarlıkTamamlandı olayının oluşturulması üzerine köle aşağıdaki eylemleri yapmalıdır:
  - DS Rxmt Zamanlayıcısını durdurmalıdır,
  - Efendi bitini (köleyi gösteren) ve Başlangıç bitini sıfıra ayarlamalıdır. DS sıra numarasını efendi tarafından belirtilen rakama ayarlamalı ve efendiye içinde bu düğümün veritabanı özetinin ilk bölümü bulunan bir Veritabanı Özet paketi göndermelidir.
- Başlangıç ve Efendi bitleri sıfırdır, paketin DS sıra numarası düğümün kendi DS sıra numarasına eşittir (alındığı gösteren) ve komşu peer'in düğüm kimliği düğümün kendi düğüm kimliğinden küçüktür. Bu

durumda bu düğüm Efendidir. PazarlıkTamamlandı olayının gerçekleştirilmesi üzerine efendi aşağıdaki eylemleri yapmalıdır:

- DS Rxmt Zamanlayıcısını durdurmalıdır,
- Alınan Veritabanı Özet paketinin içeriğini işletmelidir,
- DS sıra numarasını bir arttırmalıdır, köleye içinde bu düğümün veritabanı özetinin ilk bölümü bulunan bir Veritabanı Özet paketi göndermelidir ve DS Rxmt zamanlayıcısını tekrar başlatmalıdır.

DeğişTokuşYapılıyor : Eğer Efendi bitinin durumu, bağlantının efendi/köle durumu ile uyuşmuyorsa, DSMismatch olayı yaratılır ve paketin işlenmesi durdurulur. Aksi halde:

- Başlangıç biti ayarlı ise DSMismatch olayı yaratılır ve paketin işlenmesi durdurulur.
- Düğüm Efendi ise ve paketin DS sıra numarası düğümün kendi DS sıra numarası ile eşitse (bu paket bir sonraki sıradadır) paket kabul edilmeli ve aşağıdaki şekilde işlenmelidir:
  - DS Rxmt Zamanlayıcısı durdurulmalı,
  - Alınan Veritabanı Özet paketleri işlenmeli
  - Aşağıdaki sıra ile:

A) DS sıra numarası bir arttırılmalı

B) Düğüm önceden Veritabanı Özet paketlerinin tamamını göndermişse (örneğin, düğümün gönderdiği önceki Veritabanı Özet paketinin DahaFazla biti sıfıra ayarlı), alınan paketteki DahaFazla biti de sıfıra ayarlı ise ve PTSE Talep listesi boş değilse DeğişTokuşTamamlandı olayı gerçekleştirilmeli veya PTSE Talep listesi boş değilse ZamanUyumlamaTamamlandı olayı gerçekleştirilmelidir,

C) Aksi halde, köleye yeni bir Veritabanı özet paketi göndermeli ve DS Rxmt Zamanlayıcısını tekrar başlatmalıdır.

- Düğüm efendi ise ve paketin DS sıra numarası, düğümün DS sıra numarasından küçük ise paket kopyadır. Kopyalar efendi tarafından atılmalıdır.
- Düğüm köle ise ve paketin DS sıra numarası, düğümün DS sıra numarasından büyük ise (bu paket bir sonraki sıradadır) paket kabul edilmeli ve aşağıdaki gibi işlenmelidir (sıra önemli değildir):
  - Alınan Veritabanı özet paketinin içeriği işlenmelidir,
  - Aşağıdaki sıra ile:

D) DS sıra numarası, alınan pakette görünen DS sıra numarasına ayarlanmalıdır,

E) Efendi'ye bir Veritabanı Özet paketi gönderilmelidir,

F) Alınan paketteki DahaFazla biti sıfıra ayarlı ise ve henüz gönderilmiş Veritabanı Özet paketindeki DahaFazla biti de sıfıra ayarlı ise (örneğin, henüz gönderilmiş Veritabanı Özet paketinin içeriği boş ise), PTSE Talep listesi boş değilse DeğişTokuşTamamlandı olayı veya PTSE Talep listesi boş ise ZamanUyumlamaTamamlandı olayı gerçekleştirilmelidir.

- Düğüm köle ise, paketin DS sıra numarası, düğümün DS sıra numarasına eşitse paket kopyadır. Köle kopyalara, yolladığı son Veritabanı Özet paketini tekrarlayarak cevap vermelidir.
- Bundan başka, DSMismatch olayı gerçekleştirilmeli ve paketin işlenmesi durdurulmalıdır.

Alınan bir Veritabanı Özet paketinin içeriğinin işlenmesi:

Düğüm, almış olduğu bir sonraki sırada bulunan bir Veritabanı Özet paketini kabul ettiğinde, en son gönderilen Veritabanı Özet paketinin içeriği alınmış olarak kabul edilir ve alınmış olan Veritabanı Özet paketinin içeriği aşağıdaki şekilde işlenir:

Listedeki her PTSE için düğüm, veritabanındaki PTSE'lere, bu PTSE'nin bir örneğine sahip mi kontrol eder. Sahip değilse veya veritabanı kopyası daha eski ise aşağıdaki eylemlerden biri yapılır:

- Listedeki PTSE, bu düğümün kendi oluşturduğu PTSE'lerden biriye düğüm aşağıdakilerden birini yapmalıdır:
  - Düğümün geçerli bir PTSE örneği varsa, daha büyük sıra numarası olan daha yeni bir örneği tekrar oluşturmalı veya,
  - Topoloji veritabanına, kalan hayatını ZamanAşımıYaşına ayarlayarak yerleştirdiği PTSE'yi atmalıdır.
- Aksi taktirde, eğer listedeki PTSE'nin, PTSE KalanYaşamaSüresi ZamanAşımıYaşı ise, PTSE özetindeki PTSP ve PTSE başlık içeriği yeni veya boş içerikli güncellenmiş PTSE olarak kabul edilmelidir.
- Aksi halde PTSE, PTSE Talep listesine konur, böylece herhangi bir komşu peer'dan PTSE Talep Paketleriyle istenebilir.

**Yükleniyor veya Dolu :** Bu durumların her ikisinde de düğüm sırasıyla bir Veritabanı Özet Paketini göndermiş ve almıştır. Alınan paketler sadece kopyalar olmalıdır. Alınan diğer paketler DSMismatch olayını gerçekleştirmelidir, bu da komşuluğun PazarlıkEdiliyor durumuna geri dönmesine ve iki komşu peer'ın veri tabanlarının zamanuyumlamasını yeniden yapmasına sebep olur.

Yükleniyor veya dolu durumlarında, bir Veritabanı Özet Paketi alındığında takip edilecek prosedürler, bir sonraki sırada kabul edilen paketlerin DSMismatch olayını gerçekleştirmesinin ve bu tür paketlerin daha fazla işlenmesinin durdurulmasının dışında, DeğişTokuşYapılıyor durumuyla aynıdır. Tutarsız Efendi biti olan veya Başlangıç bit ayarı sıfıra ayarlı paketler DSMismatch olayını gerçekleştirmelidir.

#### 4.2.6 PTSE Talep Paketlerinin Gönderilmesi

DeğişTokuşYapılıyor ve Yükleniyor durumlarında PTSE Talep listesinin içinde, bu düğümün topoloji veritabanı ile komşunun topoloji veritabanı arasındaki zamanuyumlamasını yapmak için edinilmesi gereken PTSE'lerin listesi bulunur. Bu PTSE'leri talep etmek için düğüm, PTSE talep listesinden bir veya daha fazla giriş içeren bir PTSE Talep Paketi gönderir. PTSE talep paketleri sadece DeğişTokuşYapılıyor ve Yükleniyor durumlarında gönderilir. Bir düğüm, bu komşu peer'a ve/veya isteğe bağlı olarak DeğişTokuşYapılıyor ve Yükleniyor durumunda olan, veritabanı özetlerinde eksik PTSE bulunduğunu bildiren diğer komşu peer'lara da bir PTSE Talep Paketi gönderebilir.

Her komşu peer için, aynı anda, en fazla bir tane iyi PTSE Talep paketi olmalıdır. Bir düğüm cevapların akışını, her PTSE Talep paketinde bulunan uygun PTSE sayısını seçerek kontrol edebilir.

Belli bir komşu peer'a bir PTSE Talep paketinin her gönderilişinde ilgili komşu peer veri yapısındaki Talep Rxmt Zamanlayıcısı yeniden çalıştırılmalıdır. Komşu peer bu taleplere uygun PTSE'(ler) ile cevap verdiğinde, (örneğin istenenin herhangi bir alt kümesiyle) alınan PTSE'ler PTSE talep listesinden kaldırılır. Son PTSE Talep Listesinde bulunan tüm PTSE'ler alındığında, yeni bir PTSE Talep paketi aynı komşuya gönderilebilir. Eğer TalepRxmtAralığında tüm paketler alınmamışsa, eksik PTSE'leri içeren ve/veya ilgili komşu peer veri yapısındaki PTSE Talep Listesindekilerden farklı PTSE'leri içeren, yeni bir PTSE Talep listesi gönderilebilir. Eksik PTSE'ler bunun yerine DeğişTokuşYapılıyor ve Yükleniyor durumlarında olan ve veritabanı özetlerinden, eksik PTSE'lere sahip olduğu anlaşılan diğer komşu peer'lardan talep edilebilir. PTSE talep listesi boşalana kadar bu işlem devam eder.

PTSE talep listesi boşaldığında ve komşu peer durumu Yükleniyor olduğunda (örneğin, tam sıra Veritabanı Özet paketleri komşu peer'a gönderildiğinde ve buradan alındığında) YüklemeTamamlandı olayı gerçekleştirilir.

#### 4.2.7 PTSE Talep Paketlerinin Alınması

Alınan PTSE Talep paketleri, komşu peer'ın almak istediği PTSE'lerin listesini belirtir. PTSE Talep paketleri, ilgili komşu peer'ın durumu DeğişTokuşYapılıyor,

Yükleniyor ve Dolu olduğu zaman kabul edilmelidir. Tüm diğer durumlarda PTSE Talep paketleri yok sayılmalıdır.

PTSE Talep paketinde belirtilen her PTSE için, düğümün topoloji veritabanında bir örnek aranmalıdır. Talep edilen PTSE'ler PTSP'ler içinde birleştirilir (düğümün seçtiği bir tarzda) ve zamanı geldiğinde komşu peer'a gönderilir. Bu PTSE'ler ilgili komşu peer veri yapısındaki yeniden gönderme listesinde yer almamalıdır. Veritabanında bir PTSE bulunamamışsa Veritabanı Değişim sürecinde bir hata oluşmuş demektir ve BadPTSERequest olayı gerçekleştirilmelidir.

### **4.3 Topoloji Tanımı ve Dağıtımı**

#### **4.3.1 Topoloji Bilgisi**

Topoloji bilgileri içinde topoloji durum parametreleri ve düğüme ait bilgiler bulunur.

Topoloji bilgisi gelecekteki genişletilebilirliğe izin verecek şekilde belirlenir. Yani, topoloji bilgileri Tip/Uzunluk/Değer formatında kodlanır. Genişletilebilirlik, örneğin tercihli kaynak seçimi, politikayla ilgili fonksiyonlar ve işaretleşme ihtiyaçlarını karşılamak için QoS ve trafik anlaşma görüşmelerinde kullanılabilir.

##### **4.3.1.1 Topoloji Durum Parametreleri**

Topoloji durum parametresi, ya bir bağ durum parametresine ya da düğüme ait bir durum parametresine işaret eden bir terimdir. Topoloji durum parametreleri iki kategoriye ayrılır: topoloji metrik ve topoloji özellik. Topoloji metrik, verilmiş bir bağlantıyı taşımak için bir güzergahın kabul edilebilir ve/veya istenir olup olmadığına karar vermek üzere, verilen bir güzergâh üzerindeki düğümlerin ve tüm parametrelerin birleşeceği değerlere gereksinim duyan bir topoloji durum parametresidir. Topoloji özelliği, belirli bir bağın veya düğümün, belirli bir bağlantıyı taşımak için kabul edilir olup olmadığına karar vermek için ele alınan bir topoloji durum parametresidir.

Topoloji özellikleri, performans/kaynak-ilişkili topoloji özellikleri ve politika-ilişkili topoloji özellikleri olarak da ayrılabilirler. Performans/kaynak-ilişkili topoloji özellikleri, topoloji içeriğiyle ilgili kaynak kısıtı veya hücre aktarım performansının ölçüsünü temin eden bir topoloji özelliğidir. Politika-ilişkili topoloji özelliği, topoloji

içeriğinin belli bir politik kısıta uygunluk seviyesini karakterize eden bir topoloji özelliğidir. Örneğin, Kullanılabilir hücre oranı bir performans/kaynak-ilişkili topoloji özelliği iken, Kısıtlı Geçiş Bayrağı, politikayla ilişkili topoloji özelliğidir (Tablo 4.1).

Tablo 4.1 Topoloji Durum Parametreleri

| Topoloji Metrikleri                       | Topoloji Özellikleri                    |                       |
|-------------------------------------------|-----------------------------------------|-----------------------|
|                                           | Performans/Kaynak-İlgili                | Politikayla İlgili    |
| Hücre Gecikme Değişimi (CDV)              | CLP=0 için Hücre Kaybı Oranı (CLRo)     | Kısıtlı Geçiş Bayrağı |
| Maksimum Hücre Aktarım Gecikmesi (maxCTD) | CLP=0+1 için Hücre Kaybı Oranı (CLRo+1) |                       |
| Yönetimsel Ağırlık (AW)                   | Maksimum Hücre Oranı (maxCR)            |                       |
|                                           | Mevcut Hücre Marjini (AvCR)             |                       |
|                                           | Hücre Oran Marjini (CRM)                |                       |
|                                           | Değişim Faktörü (VF)                    |                       |
|                                           | Kısıtlı Dallanma Bayrağı                |                       |

#### Bağ Durum Parametreleri

Bağ durum parametresi, bir bağın görünüşünü ve özelliğini gösteren bilgiyi sağlar. Bir ATM ağındaki bağlar iki-yönlü olduğundan, bağ durum parametresi tarif ettiği bağın yönünü dahili olarak bulundurur. PNNI hiyerarşik yapısında, bir düğüme bağlı bağ, yatay bağ ya da üst bağ olabilir. Yatay bağ, aynı grup içindeki iki düğüm arasındaki bağıdır. Bir Üst bağ, bir peer gruptaki sınır düğüm ile bu peer grubun dışındaki bir üst seviye üst düğüm arasındaki bağlanabilirliği temsil eder.

#### Düğüme Ait Durum Parametreleri

Düğüme ait durum parametreleri, bir düğümün görünüşünü ve özelliğini içeren bilgiyi sağlar. Düğümsel durum parametreleri PNNI kompleks düğüm gösterimini inşa etmek için kullanılır. Bir düğümsel durum parametresi yöne bağlıdır ve yön mantıksal düğümün bir çift girdi ve çıktı Kapı Kimlikleri tarafından belirlenir.

### **Kaynak Kullanılabilirliği Bilgi Grubu**

Kaynak Kullanılabilirliği Bilgi Grubu (RAIG) içinde, topoloji durum parametreleri değerlerini düğümlere, bağlara ve ulaşılabilir adreslere bağlamak için kullanılan bilgiler bulunur.

### **Kaynak Kullanılabilirliği Bilgi Grubu Bayrakları**

Bu bayraklar, PNNI protokol mesajları içindeki RAIG'lerde bulunan topoloji durum parametrelerinin yorumlanmasını etkiler.

RAIG'nin ilk 5 biti, topoloji durum parametrelerinin RAIG başvurusunda hangi hizmet kategorilerini içerdiğini belirtir. Belirtilen her hizmet kategorisi için 1 bit bulunur. Bu sayede, bir grup topoloji durum parametresi bir veya daha fazla hizmete başvurulabilir. Farklı hizmet kategorileri için topoloji durum parametreleri aynı değilse, her biri topoloji durum parametresi taşıyan birçok RAIG ilan edilir; bu RAIG'lar bütünü anlatan tek bir bilgi grubu içinde bulunurlar. Her hizmet kategorisi için en fazla ilan edilen bir küme topoloji durum parametresi bulunur.

### **Hücre Gecikme Değişimi (CDV)**

CDV, CBR ve Gerçek Zamanlı VBR hizmet kategorileri için gereken bir topoloji metriğidir. CDV, Gerçek Zamanlı Olmayan VBR, ABR ve UBR hizmet kategorilerinde kullanılamaz.

### **Maksimum Hücre Aktarım Gecikmesi (maxCTD)**

MaxCTD, bağ veya düğümden geçişteki, sabit gecikme içeriklerinin ve CDV'nin toplamıdır. MaxCTD, CBR, gerçek zamanlı VBR ve gerçek zamanlı olmayan VBR hizmet kategorileri için gereken bir topoloji durum parametresidir. UBR ve ABR hizmet kategorilerinde uygulanamaz.

### **Yönetimsel Ağırlık (AW)**

Yönetimsel ağırlık, ağ operatörü tarafından belirlenen bir değerdir. Ağ operatörünce önemli bir sebep için, bir bağ veya düğümün göreceli kullanım isteğini belirtmek için kullanılır. Yönetimsel ağırlık tüm hizmet kategorileri için gereken bir metriktir. Varsayılan değeri, DefaultAdminWeight olan boyutsuz bir büyüklüktür. AW'nin yüksek bir değeri, düşük değerli bir bağdan daha az kullanılmak istenen bir düğümü

veya bağı anlatır. Bir güzergahın yönetsel ağırlığı, bağların yönetsel ağırlıkları ile güzergahta bulunan düğümlerin toplamıdır.

#### **CLP=0 İçin Hücre Kayıp Oranı (CLR<sub>0</sub>)**

CLR, bağı veya düğümün sonuna ulaşamayan hücre sayısının, düğüm veya bağdan gönderilen toplam hücre sayısına oranıdır. (CLP<sub>0</sub>), CBR, gerçek zamanlı VBR ve gerçek zamanlı olmayan VBR hizmet kategorileri için gereken bir topoloji özelliğidir. ABR ve UBR hizmet kategorilerinde kullanılamaz.

#### **Maksimum Hücre Oranı (maxCR)**

MaxCR, bağlantılar tarafından kullanılabilir olan maksimum kapasiteyi gösterir. MaxCR, ABR ve UBR hizmet kategorileri için gereken bir topoloji özelliğidir. CBR, gerçek zamanlı VBR ve gerçek zamanlı olmayan VBR hizmet kategorileri için opsiyoneldir.

#### **Kullanılabilir hücre oranı (AvCR)**

AvCR, CBR, Gerçek zamanlı VBR ve Gerçek zamanlı olmayan VBR hizmet kategorileri için etkin mevcut kapasite ölçüsüdür. ABR hizmet kategorisi için AvCR, minimum hücre oranı (MCR) rezervasyonu için mevcut kapasite ölçüsüdür.

AvCR, CBR, gerçek zamanlı VBR, gerçek zamanlı olmayan VBR ve ABR hizmet kategorileri için gereken bir topoloji özelliğidir. UBR hizmet kategorisinde kullanılamaz. Saniye başına düşen hücre birimleri ile ifade edilir.

#### **Değişim Faktörü (VF)**

VF, varolan tüm bağlantıların hücre oranlarının toplamının değişimiyle normalleştirilen hücre oran marjininin karesinin göreceli ölçüsüdür.

VF, gerçek zamanlı VBR ve gerçek zamanlı olmayan VBR için opsiyonel bir topoloji özelliğidir. CBR, ABR ve UBR hizmet kategorilerinde kullanılamayan boyutsuz bir orandır.

#### **4.3.1.2 Dügüme Ait Bilgiler**

Dügüme ait bilgiler düğümü tanımlar, kaynak kullanılabilirliği bilgisini ve ulaşılabilirlik bilgisini içermezler. Dügüme ait bilgiler, aşağıdakileri içerir:

- D ğ m n ATM U  Sistem Adresi
- Liderlik  ncelięi
- D ğ me ait bilgi bayrakları
- Tercih edilen peer grup liderinin d ğ m Kimlięi
- Sonraki y ksek seviye baęlama bilgisi (Peer Grup Liderinde bulunur).

#### **D ğ m n ATM U  Sistem Adresi**

D ğ me ulařmak i in kullanılabilecek ATM adreslerini temin eder. Bu alan, d ğ m n, D ğ m Kimlięinden farklıdır.

#### **Liderlik  ncelięi**

Bu alan, bu d ğ m n Peer Grup Lideri olarak kullanılma  ncelięini belirtir ve PGL se iminde kullanılır.

#### **D ğ me Ait Bilgi Bayrakları**

Beř bayrak tanımlanmıřtır: “ben liderim” bayraęı, Kısıtlı Ge iř bayraęı, D ğ msel G sterim bayraęı, Kısıtlı Dallanma bayraęı ve PGL Se imi İ in Non-transit bayraęı. Bu alanın geri kalanı gelecek PNNI s r mlerin kullanımı i in ayrılmıřtır.

“Ben Liderim” bayraęı, peer grup lideri olduęunu ilan etmek i in, bir d ğ m tarafından 1’e ayarlanır. Bu, PGL se imi sırasında kullanılır.

Bazı durumlarda bir PNNI d ğ m n n, belirli kısıtlı trafik k meleri dıřında, ge iř ama lı olarak kullanılmasına izin verilmez. Kısıtlı Ge iř bayraęı, d ğ m n kısıtlı olup olmadıęını belirlemek amacıyla, d ğ m durum parametrelerinde verilir. Eęer d ğ m kısıtlıysa, d ğ m durum parametrelerinin bařka bir yerinde a ık a tanımlanan veya kısıtlamayı bitiren bilgi olmadıęı s rece, d ğ m ge iř ama lı olarak kullanılmaz. Kısıtlı Ge iř bayraęı bir politika-iliřkili topoloji  zellięidir.

Bir d ğ m n Kısıtlı Ge iři Bayraęı 1’e ayarlanmışsa, bu d ğ m bir DTL’nin ilk veya son d ğ m  olarak kullanılır.

Kısıtlı Geçiş bayrağı, düğümün geçişe kapalı olduğunu anlatmak için tek başına kullanılabilir. Ayrıca ek bilgiler kullanılarak, düğümün sadece kısıtlı şartlar altında geçiş için kullanılabileceğini de anlatabilir. Örneğin, gelecekte yeni bir PNNI bilgi grubunun tanımlanacağı ve bu yeni bilgi grubunun, düğümün geçiş için uygun olup olmadığına karar vermek için kullanılabileceği farz edilirse, bu düğüm Kısıtlı Geçiş olarak işaretlenecektir. Bu durumda, eski bir düğüm, (yeni tanımlanan bilgi grubunu bilmeyen düğüm) Kısıtlı Geçiş olarak işaretlenen bu düğümü, direkt ulaşılabilir sistemlere (bu düğüm yoluyla) ulaşmak için kullanır. Yeni bir düğüm bilgi grubunu yorumlayacak ve Kısıtlı Geçiş bayrağını önemseyip önemsemeyeceğine karar vermek için yeni bilgi grubunun değerlerini kullanabilecektir.

Düğümsel Gösterim biti, basit düğüm gösterimiyle mi, yoksa kompleks düğüm gösterimiyle mi temsil edildiğini ilan etmek için, bir düğüm tarafından kullanılır. Bu bitin değeri, Basit düğüm gösterimini belirtmek için 1'e ayarlanmışsa, bu düğümünden hiçbir düğüme ait durum parametre PTSE'si, yön hesaplamalarında kullanılmaz; bunun yerine, düğüm tek bir nokta olarak temsil edilir ve ilgili tüm topoloji durum parametreleri sadece ilgili yatay bağlarla, üst bağlarla, dış ulaşılabilir adreslerle ve düğüme bağlı geçiş ağları ile verilir.

Kısıtlı Dallanma bayrağı, bir düğümün tek noktadan-çok noktaya trafiği dallara ayırabilip ayıramayacağını belirtmek için kullanılır. Kısıtlı Dallanma bayrağı kaynak-ilişkili bir topoloji özelliğidir.

PGL Seçimi için geçiş yok bayrağı, PGL seçim algoritmasında bağlanabilirlik hesaplanırken bir düğümün yok sayılması gerektiğinde, bu düğüm tarafından ayarlanır.

### **Tercih Edilen Peer Grup Lider Düğüm Kimliği**

Bu alan, düğümün PGL tercihi olarak karar verdiği düğümün, düğüm Kimliğini belirtir. PGL seçiminde kullanılır.

### **Sonraki Üst Seviye Birleştirme Bilgisi**

Bu alanda, baba peer grupta bu peer grubu temsil eden, üst seviye birleştirme bilgisi hakkında peer gruptaki tüm düğümlere haber vermek için, baba LGN'ye PGL'ler

tarafından gönderilen bilgi grubu bulunur. Hiyerarşi seviyelerinden geçen yolların hesaplanması için kullanılır.

#### **Baba Mantıksal Grup Düğüm Kimliği**

Bu alan, baba LGN'nin düğüm kimliğini içerir.

#### **Baba Mantıksal grup Düğümünün ATM Uç Sistem Adresi**

Bu alanda, baba LGN'nin ATM Uç Sistem Adresi bulunur.

#### **Baba Peer Grup Kimliği**

Bu alanda, baba LGN'nin ait olduğu peer grubun peer grup kimliği bulunur.

#### **4.3.1.3 Ulaşılabilirlik Bilgisi**

Ulaşılabilirlik Bilgisi, ulaşılabilir adresleri PNNI hiyerarşisi içindeki düğümlere bağlayan topoloji bilgisidir.

Dahili ve harici ulaşılabilirlik bilgileri, kaynaklarına göre mantıksal olarak ayırt edilirler. Harici ulaşılabilirlik, PNNI yönlendirme alanının dışındaki protokol değişimlerinden türer. Dahili ulaşılabilirlik ise, PNNI yönlendirme alanının içindeki ulaşılabilirliğin yerel bilgisini temsil eder. Bu ayrımın başlıca önemi, harici ulaşılabilirlik bilgisinin diğer yönlendirme protokollerinde veya yönlendirme alanlarında ilan edilmeyecek oluşudur. Diğer yönlendirme protokollerine veya alanlarına ilan edilenlere uygun harici veya dahili ulaşılabilirlik yaratmak için manuel konfigürasyon kullanılabilir.

#### **4.3.2 PTSP'ler ve PTSE'ler**

Topoloji bilgisi, peer gruptaki tüm düğümlere güvenilir şekilde dağıtılmıştır. Bu bilgi, peer grubun topolojisini, peer gruptaki tüm düğüm ve bağları, her düğümün ulaşabileceği hedefleri dahil ederek anlatır.

##### **4.3.2.1 PTSP'ler**

PTSP, PNNI topoloji durum bilgisini komşu peer'a göndermek için kullanılan pakettir. PTSP'nin içinde topolojinin kişisel özelliklerini anlatan bir veya daha fazla

PTSE bulunur. Her PTSP, tek bir başlatıcıdan tek bir kapsam için bilgi taşır. Bunlar aşağıdaki iki PTSP başlık alanında taşınır:

- Başlatan Düğümün Kimliği

Bu PTSP içinde bulunan tüm PTSE'leri başlatan düğümün, düğüm belirteci

- Başlatan Düğümün Peer Grup Kimliği

Bu PTSP içinde bulunan tüm PTSE'leri başlatan düğümün peer grup belirteci. Bu PTSP'de bulunan PTSE'lerin taşkın mekanizmasıyla ne kadar süre dağıtılacağını belirtir. Bu süre tanıtılan peer grubunun tüm çocuk peer gruplarını mantıksal olarak kapsar.

PTSE'lerin, PTSP'ler içinde gruplanması farklı şekillerde yapılabilir. Gönderenin, PTSE'leri, komşu peer'lardan alındıkları şekildeki PTSP'ler içinde gruplanmış olarak göndermesi gerekmez.

#### 4.3.2.2 PTSE'ler

PTSE'ler taşkın ve yeniden gönderme birimleridir. Bir düğüm, o anki işletim karakteristiklerini anlatmak için bir veya daha fazla PTSE yaratır (örneğin: aktif bağlar, kullanılabilir band genişliği, vb.) ve bunları taşkın yoluyla dağıtır.

#### PTSE Başlık İçerikleri

Her PTSE başlığı aşağıdaki maddeleri içerir:

- PTSE Belirteci

Bir düğüm, her biri düğümün farklı çevre parçalarını tanımlayan (örneğin, her biri farklı bağ kümelerini anlatabilir) birden fazla PTSE başlatırsa, PTSE'ler PTSE belirteci ile ayrıştırılır. Bu işaretli 32-bitlik bir rakamdır, atanması tamamen başlatan düğüme bağlıdır.

- PTSE Sıra Numarası

Aynı PTSE'nin birden çok örneği aynı anda mevcutsa, PTSE sıra numarası hangi örneğin daha “yeni” olduğuna karar verir. PTSE Sıra Numarası 32-bitlik işaretsiz bir rakamdır.

- PTSE Denetim toplamı

a. Bir PTSE'nin taşkın sırasında ya da düğümün topoloji durum veritabanında tutulurken, bozulup bozulmadığını belirtir, ve

b. Belli istisnai durumlarda, iki PTSE örneğinden hangisinin daha yeni olduğuna karar vermek için kullanılır.

- PTSE Kalan Yaşam Süresi

PTSE'ye geçersiz sayılmadan önce, kalan saniye sayısıdır. PTSE, Kalan Yaşam Süresine,

a. PTSE, düğümün topoloji veritabanında tutulurken ve

b. taşkın sırasında karar verir.

Eğer PTSE Kalan Yaşam Süresi, ZamanAşımıYaşına ulaşırsa, taşkın yeniden yapılır ve peer grubun topoloji veritabanından kaldırılır. Bu alanın başlangıç değeri, PTSEYenilemeAralığı'nın, PTSEYaşamSüresiFaktörü ile çarpımıdır.

- PTSE Tipi

PTSE Tipi alanı, hangi kısıtlı bilgi gruplarının PTSE içinde görünmesine izin verileceğini belirtir.

### **PTSE Tanımlaması ve PTSE Örnek Tanımlaması**

Taşkın prosedürlerini uygulamak için, her PTSE ve verilen PTSE'nin her örneği tek tek tanımlanmalıdır. Bir PTSE'yi tanımlamak için kullanılan bilgiler:

- Başlatan Düğüm Kimliği
- PTSE Belirteci' dir.

Bir PTSE örneğini, aynı PTSE'nin başka bir örneğinden ayırmak için kullanılan ek bilgiler ise:

- PTSE Sıra Numarası
- PTSE Kalan Yaşam Süresi
- PTSE Denetim toplamıdır.

PTSE'yi içeren PTSP'de taşınan, Başlatan Düğüm Kimliği dışında, tanımlayıcı verilerin çoğu PTSE başlığında taşınır.

### **PTSE Örneklerinin Karşılaştırılması**

Aynı Başlatan Düğüm Kimliği ve PTSE Belirtecine sahip fakat PTSE Sıra Numaraları farklı olan iki PTSE, aynı PTSE'nin farklı örnekleridir. Örneğin, bir PTSE diğerinin güncellenmiş hali olabilir.

Aynı PTSE'nin iki örneği aynı anda varolduğunda, farklı örnekler olup olmadıklarını anlamak için kıyaslama yapılır. Farklılarsa, böylece hangisinin “daha yeni” olduğu anlaşılır. Daha yeni PTSE her zaman daha eski PTSE'den önceliklidir. Bu kıyaslama aşağıdaki basamakları içerir:

- 1) 32-bit işaretli tamsayı olarak kıyaslandığında daha büyük PTSE Sıra Numarasına sahip PTSE Örneği, daha yeni olarak kabul edilir.

Aksi halde (örneğin her iki kopyanın da PTSE Sıra Numaraları aynıysa),

- 2) PTSE kopyalarının birinin PTSE Kalan Yaşam Süresi, ZamanAşımıYaşına eşitse ve diğerinki değilse, PTSE Kalan Yaşam Süresi, ZamanAşımıYaşına eşit olan kopya daha yeni olarak kabul edilir.

Aksi halde (örneğin her iki kopya da tükenmişse veya her ikisi de tükenmemişse),

- 3) İşaretsiz sayılar olarak kıyaslandığında PTSE Denetim toplamı değeri daha büyük olan PTSE kopyası, daha yeni olarak kabul edilir. (Not: İki kopya da farklı PTSE Denetim toplamlarına sahipse, farklı içerikleri vardır, yani farklı örneklerdir. Ama, bu noktada hangisinin “daha yeni” olduğuna karar vermek

imkansızdır. Eğer seçilen PTSE kopyası gerçekten de en yeni olan değilse, PTSE'nin başlatıcısı, PTSE'yi yeniden yaratmaya son verir.)

Aksi halde her iki kopya da aynı Denetim toplamı'a sahiptir ve her iki kopyanın da aynı PTSE örneği olduğu kabul edilir.

#### **4.3.3 Taşkın Yöntemi**

PNNI taşkın algoritması, bir peer grup boyunca PTSE'lerin güvenilir olarak dağıtılmasını sağlar. Peer gruptaki her düğümün zamanuyumlaması yapılmış veritabanına sahip olmasını garanti eder (örneğin PTSE'lerin toplanması).

DeğişTokuşYapılıyor, Yükleniyor veya Dolu durumlarındaki tüm komşuluklar, taşkın prosedürleri tarafından kullanılır ve tüm PNNI yönlendirme paketlerinin gönderme ve alma işlemlerini yapabilmelidir.

Esas olarak, taşkın prosedürü şu şekildedir: PTSE'ler, PTSP'ler içine kapatılmışlardır. Bir PTSP alındığında PTSE içeriği incelenir. Her PTSE, bir "Alındı Paketi" içinde kendi PTSE tanımlayıcı bilgisi ile onaylanır, bu onay paketi sonra komşu peer'a geri gönderilir. Buna ek olarak eğer PTSE, düğümün o anki kopyasından daha yeni ise, PTSE topoloji veritabanına yerleştirilir ve tüm diğer komşu peer'lara taşkın edilir. PTSE'lerin bu komşu peer'lara gönderilmiş olduğu hatırlanır ve alındı alınana kadar tekrar tekrar gönderilmeye devam edilir.

##### **4.3.3.1 PTSP'lerin Gönderilmesi**

PNNI düğümleri, PNNI Topoloji Durum Paketlerindeki (PTSP'ler) bağ durum bilgilerini değiştirirler.

Düğüm, aşağıdaki olaylara tepki olarak PTSP'ler geliştirip gönderirler:

- yeni bir PTSE'nin oluşması
- oluşan yeni bir PTSE örneğinin tekrar oluşması
- veritabanına, diğer peer'lara taşkın edilmek üzere alınan, kendi kendine oluşmamış, yeni PTSE örneklerinin yüklenmesi

- PTSE yeniden gönderme zamanlayıcısının, Peer Yeniden Gönderme Listesinde onaylanmamış PTSE'ler olduğu halde zaman aşımına uğraması
- PTSE talebine tepki olarak
- PTSE kalan yaşam süresinin tükenmesi

Bir düğümün bir PTSP geliştirip göndermesine sebep olan olaya bakılmaksızın, yapılanlar hep aynıdır. PTSE'lerin herhangi bir paketlenme işlemi ilk olarak yapılır. PNNI protokolünde, gönderimin tek bir düğüm tarafından yapılması için, tüm PTSE'ler tek bir PTSP'de paketlenir.

Genelde, mümkünse PTSE'nin tek bir PTSP'de paketlenmesi tavsiye edilir. Ama bir düğüm, PTSP'nin gönderileceği bağda izin verilen maksimum paket boyutundan daha büyük PTSP'ler oluşturmamalıdır. Ayrıca düğüm kullandığı bağ için geçerli olan trafik anlaşmasını bozmamalıdır. Yani, verilen bir komşuya giden, verilen bağda dizilecek tüm PTSP'lerin toplam sayısı ve boyutu, gönderen düğüm tarafından gözlemlenip kontrol edilir. Eğer gönderen düğüm, bağın veya trafik anlaşmasının sınırları içinde kalmak için hemen bir PTSP göndermediği noktaya ulaşırsa, durum kaydedilmelidir, böylece gerekli olan PTSP güncellemesi, müsait bir durumda gönderilir.

Herhangi bir PTSE yeniden gönderimi için farklı şekilde PTSP'lere paketlenabilir. Düğümlerin, diğer bir düğümün PTSE'lerini farklı bir PTSP paketi içinde düzenlemelerine izin verilir. Bu, bir düğümün, başka bir düğüm tarafından oluşturulmuş PTSE içine bilgileri yeniden düzenleyemeyeceği konusundaki kısıtlama ile tezat oluşturur.

#### **4.3.3.2 Bir PTSP'nin Alınması**

Komşu peer'dan bir PTSP alındığında, aşağıdaki adımlar uygulanarak, içerdiği tüm PTSE'ler incelenir:

1) Eğer PTSE Kalan Yaşam Süresi, ZamanAşımıYaşından farklıysa, PTSE Denetim toplamı onaylanır. Eğer Denetim toplamının geçersiz olduğuna karar verilirse PTSE'nin işlenmesi biter. Aksi halde,

2) Eğer alınan bağı komşu peer durumu DeğişTokuşYapılıyor veya Yükleniyor ise ve alıcı bağı PTSE talep listesinde PTSE'nin bir örneği varsa, alınan PTSE talep listesindeki örnekle karşılaştırılır:

a) Alınan PTSE, talep listesindekinden daha eski ise veritabanı değişim sürecinde bir hata oluşmuştur. Bu durumda, BadPTSERequest olayı gerçekleştirilerek, komşu peer'la veritabanı değişim işlemi tekrar başlatılıp PTSP'nin işlenmesi durdurulur.

b) Aksi halde, PTSE alıcı bağı PTSE Talep listesinden silinir ve bu PTSE'nin işlenmesine devam edilir.

3) Düğümün topoloji veritabanında alınan PTSE'den daha yeni bir PTSE örneği varsa ve

a) Bağı Peer Yeniden Gönderme Listesinde bu PTSE bulunmuşsa, PTSE'nin alındısı derhal onaylanır ve PTSE'nin işlenmesi tamamlanır.

b) Bağı Peer Yeniden Gönderme Listesinde bu PTSE bulunmamışsa, veritabanı kopyası bir PTSP içine paketlenip komşu peer'a taşkın edilir. PTSP içindeki PTSE kopyasının PTSE Kalan Yaşam Süresi 1 azaltılır (PTSE Kalan Yaşam Süresi zaten ZamanAşımıYaşına eşit değilse); bu şekilde oluşabilecek herhangi bir taşkın döngüsü engellenmiş olur. Ayrıca, PTSE komşunun Peer Yeniden Gönderme Listesine eklenir, böylece onay alınana kadar PTSE tekrar tekrar gönderilir. Aksi halde,

4) Veritabanında bir örnek varsa ve alınan PTSE'deki örnekle aynıysa, o zaman:

a) Eğer PTSE, alınan bağı Peer Yeniden Gönderme Listesinde mevcutsa, taşkın edilen PTSE'ye alındısı varmış gözüyle bakılır. Bu durumda, PTSE alınan bağı Peer Yeniden Gönderme Listesinden kaldırılır ve PTSE'nin işlenmesi biter. Aksi halde,

b) PTSE'nin alındısı, bir PTSE Alındı Paketi yollanarak derhal onaylanır ve PTSE'nin işlenmesi tamamlanır.

Aksi halde,

5) Eğer veritabanı kopyası yoksa, ve alınan PTSE’de, PTSE Kalan Yaşam Süresi ZamanAşımıYaşına eşitse, PTSE’nin alındısı bir PTSE Alındı Paketi yollanarak derhal onaylanır ve:

a) Eğer alıcı bağı komşu peer durumu DeğişTokuşYapılıyor veya Yükleniyor ise, zaman aşımına uğramış PTSE, alıcı düğümün topoloji veritabanına yerleştirilir ve PTSE’nin işlenmesi tamamlanır. Bu, belli yarış koşulları altında BadPTSERequest olayından kaçınmak için yapılır. Aksi halde,

b) PTSE’nin işlenmesi tamamlanır.

Aksi halde,

6) Ya bir veritabanı kopyası yoktur ve alınan PTSE ZamanAşımıYaşında değildir, ya da alınan PTSE örneği veritabanı kopyasındakinden daha yenidir. Bu durumda, aşağıdaki işlemler yapılmalıdır:

a) Eğer Başlatan Düğüm Kimliği, alıcı düğüm tarafından listelenmişse, (örneğin alıcı düğümün Düğüm Kimliği ile eşitse) o zaman:

(i) Eğer alıcı düğümün geçerli bir PTSE örneği varsa, PTSE’yi, alınan PTSE’deki sıra numarasından daha büyük bir sıra numarasıyla, yeniden başlatır ve PTSE’nin işlenmesi tamamlanır.

(ii) Alınan PTSE’nin Kalan Yaşam Süresi, ZamanAşımıYaşına ayarlanarak, tükenen PTSE topoloji veritabanına yerleştirilir ve PTSE taşkını başlatılır. Bunlardan sonra PTSE’nin işlenmesi tamamlanır.

Aksi halde,

b) PTSE, alıcı bağı Peer Gecikmiş Alındı Listesine eklenir. Buna ek olarak,

c) PTSE, alıcı düğümün topoloji veritabanına yerleştirilir. Bu süreçte, önceki veritabanı kopyası (varsa), komşu peer’ın tüm Peer Yeniden Gönderme Listesinden ve Peer Gecikmiş Alındı Listesinden silinmelidir (alıcı bağı uygun olanlar da dahil).

d) PTSE, tüm komşu peer'ların bir alt kümesine taşkın edilir. Her komşu peer için aşağıdaki işlemler tekrar edilir:

(i) Eğer yeni PTSE bu komşudan alınmışsa, bir sonraki komşu peer incelenir. Aksi halde,

(ii) Komşu peer DeğişTokuşYapılıyor durumundan daha düşük bir durumdaysa, taşkına katılmaz ve bir sonraki komşu peer incelenir. Aksi halde,

(iii) Eğer komşu peer durumu DeğişTokuşYapılıyor veya Yükleniyor ise, bu komşuluk ile ilgili PTSE Talep listesi incelenir. Eğer yeni PTSE'nin bir örneği listede varsa, yeni PTSE komşu peer'ın kopyası ile karşılaştırılır:

- Yeni PTSE daha eski ise sonraki komşu peer incelenir.
- İki kopya aynı örneğe PTSE, PTSE talep listesinden silinir, sonraki komşu peer incelenir.
- Yeni PTSE daha günceldir. PTSE, PTSE talep listesinden silinir ve (iv)'de anlatıldığı şekilde komşu peer'a taşkın edilir.

Aksi halde,

(iv) PTSE, bir PTSP içine paketlenip komşu peer'a taşkın edilir. PTSP'deki PTSE kopyasının PTSE Kalan Yaşam Süresi 1 azaltılır (PTSE Kalan Yaşam Süresi zaten ZamanAşımıYaşına eşit değilse). Ayrıca, PTSE komşu peer'ın Peer Yeniden Gönderme Listesine eklenir, böylece onaylana kadar tekrar tekrar gönderilir.

#### 4.3.3.3 Peer Yeniden Gönderme Listesinin İşletilmesi

Bir düğüm periyodik olarak her komşu peer'ının Peer Yeniden Gönderme Listesini, yeniden gönderilmesi gereken PTSE'lerin belirlenmesi için incelemelidir. PTSEYenidenGöndermeAralık saniyesinden önce gönderilen PTSE'ler bir PTSP içinde paketlenip komşu peer'a gönderilir. Düğümün topoloji veritabanında o an bulunan PTSE sürümü bu amaç için kullanılır.

#### **4.3.3.4 PTSE Alındılarının Gönderilmesi**

Bölüm 4.3.3.2’de anlatılan prosedürlere göre, bazı PTSE’lerin alındısı derhal gönderilirken bazılarının alındısı gecikebilir. Peer Gecikmiş Alındı listesi, PeerGecikmişAlındıAralığı adı verilen belli bir gecikmeden sonra gönderilecek PTSE onaylarını barındırır. Bu liste, her PeerGecikmişAlındıAralığı saniyesinde taranır ve boş değilse, PTSE tanımlayıcı bilgi maddelerini içeren bir paket yaratılıp gönderilir. Onaylanmış PTSE’ler, Peer Gecikmiş Alındı listesinden silinir. Peer Gecikmiş Alındı listesindeki PTSE alındıları, acil alındılarla birlikte gönderilebilir.

#### **4.3.3.5 Alındı Paketlerinin Alınması**

Bir Alındı Paketinde, PTSE tanımlayan bilgi maddelerinin listesi bulunur. PTSP’lere cevap olarak gönderilirler.

Komşu peer’dan bir Alındı Paketi alındığında, bilgi maddelerini belirten PTSE listesi incelenir. Her PTSE tanımlayan bilgi maddesi için, eğer komşu peer’ın Peer Yeniden Gönderme Listesinde tamamen aynı PTSE örneği varsa; PTSE, Peer Yeniden Gönderme Listesinden kaldırılır.

#### **4.3.3.6 Yeni PTSE’nin veya Yeni PTSE Örneğinin Başlatılması**

Bir düğüm, kendi kendine oluşan PTSE’lerinden birini başlattığında veya güncellediğinde, bir PTSE’nin taşkını başlatır.

Düğüm bir PTSE başlattığında veya daha önceden oluşturduğu PTSE’yi güncellediğinde aşağıdaki basamakları takip eder:

- I. PTSE’nin PTSE Sıra Numarası için bir değer seçer. Eğer:
  - A. PTSE’nin ilk başlatılışı ise, PTSE Sıra Numarası, Başlangıç Sıra Numarasına ayarlanır.
  - B. Taşkın sırasında alınan kendi kendine oluşan bir PTSE’ye cevap ise, alınan PTSE’de listelenmiş numaradan sonraki PTSE Sıra Numarası seçilir.
  - C. Varolan bir PTSE’nin güncellemesi ise, o anki veritabanı kopyasında listelenmiş numaradan sonraki PTSE Sıra Numarası seçilir.

PTSE Sıra Numarası uzayının çabuk tükenmesini engellemek için bir düğüme, belli bir PTSE'yi her MinPTSEAralık saniyesinde bir kereden fazla güncellemesine izin verilmez. MinPTSEAralığı için 1 saniye varsayılan değeri ile, hatalar oluşmadığı zaman, PTSE Sıra Numarası boşluğu en kısa olarak 136 yılda tükenebilir.

- II. Düğüm, düğümün çevresini anlatan bir PTSE oluşturur. PTSE'nin en son oluştuğu zamandan bu yana önemli bir değişiklik olsa da olmasa da, bir PTSE her yeniden başlatılışında düğümün en son halini yansıtmalıdır.
- III. Düğüm, PTSE Denetim toplamını hesaplar ve yerleştirir.
- IV. Düğüm, PTSEYenilemeAralığını PTSEYaşamSüresiFaktörü ile çarparak kalan yaşam süresini hesaplar ve sonucu PTSE Kalan Yaşam Süresi olarak yerleştirir.
- V. PTSE, düğümün topoloji veritabanına yerleştirilmiştir. PTSE'nin önceki örneği (varsa), tüm komşu peer'ların Peer Yeniden Gönderme Listesinden ve Peer Gecikmiş Alındı Listesinden silinir.
- VI. DeğişTokuşYapılıyor veya daha ileri durumlardaki tüm komşu peer'lar için, PTSE bir PTSP içine paketlenip komşu peer'a taşkın edilir. Ayrıca PTSE, komşu peer'ların Peer Yeniden Gönderme Listesine eklenir.

#### 4.3.3.7 PTSE'nin Yaşam Süresinin Tükenmesi

Bir PTSE yaşam süresi iki şekilde tükenir:

- 1) Düğüm kendi kendine oluşan PTSE'lerden birini erkenden yaşlandırır. Buna sıklıkla, PTSE'deki bilgilerin geçersiz olması sebep olur.
- 2) Düğümün topoloji veritabanındaki PTSE'lerden biri normal, şartlarla yaşlanır.

Bazen bir düğüm kendi kendine oluşmuş bir PTSE'yi *doğal olarak yaşlanmadan* kaldırmak ister. Düğüm, PTSE Kalan Yaşam Zamanını, ZamanAşımıYaşına ayarlar. Buna “erken yaşlanma” denir. Bir düğümün sadece kendi ürettiği PTSE'leri “erkenden yaşlandırmasına” izin verilir (örneğin, Başlatan Düğüm Kimliği, düğümün kendi Belirtecine eşit olanlar).

Bir PTSE'nin kalan yaşam süresinin her ZamanAşımıYaşına eşit oluşunda (ya erken ya da doğal yaşlanmaya bağlı olarak) PTSE harekete geçirilir. PTSE'yi harekete geçirmek için aşağıdaki işlemler uygulanır:

- I. PTSE, tüm komşu peer'ların Peer Yeniden Gönderme Listesinden ve Peer Gecikmiş Alındı listesinden silinir.
- II. PTSE, Kalan Yaşam Süresini, ZamanAşımıYaşına ayarlar.
- III. PTSE, değiştiriliyor veya DeğişTokuşYapılıyor ya da daha ileri durumlardaki tüm komşu peer'lara taşkın edilir. Tükenen PTSE, komşu peer'ın Peer Yeniden Gönderme Listesine eklenir.
- IV. Bölüm 4.3.3.8'de anlatılan koşullar yerine getirildiğinde tükenen PTSE topoloji veritabanından kaldırılır.

#### **4.3.3.8 PTSE'lerin Topoloji Veritabanından Kaldırılması**

Bir PTSE topoloji veritabanından sadece ve sadece aşağıdaki durumlarla karşılaşıldığında kaldırılmalıdır:

- 1) PTSE'nin, PTSE Kalan Yaşam Süresi, ZamanAşımıYaşına eşit olduğunda,
- 2) PTSE, artık düğümün hiçbir Peer Yeniden Gönderme Listesinde veya düğümün Peer Gecikmiş Alındı Listesinde yer almadığında,
- 3) Düğümün hiçbir komşu peer'ı, DeğişTokuşYapılıyor veya Yükleniyor durumunda değilse.

#### **4.3.4 Topoloji Veritabanındaki PTSE'lerin Yaşlandırılması**

PNNI düğümleri, topoloji veri tabanlarında toplanan PTSE'lerin yaşlarını gözlemler. Yaşlanma fonksiyonu, topoloji veritabanına en son PTSE girişinden bu yana geçen zamanı yansıtmak için PTSE Kalan Yaşam Süresini ayarlar. Yaşlanma fonksiyonu ayrıca, kendi kendine başlayan PTSE'lerin ne zaman yenileneceğini ve kendi kendine başlamamış olan PTSE'lerin ne zaman harekete geçirileceğini de belirler (örneğin, ZamanAşımıYaşına ulaştıkları için).

#### **4.3.4.1 PTSE’lerin Kalan Yaşam Süresinin Hesaplanması**

PTSE’nin kalan yaşam süresi, PTSE’nin veritabanına yerleştirildiğinde kalan yaşam süresinin ne kadar olduğunu ve o zamandan bu yana ne kadar zaman geçtiğine bağlı olarak belirlenir. Eğer baştaki kalan yaşam süresi geçen zamana eşitse veya daha azsa, kalan yaşam süresi ZamanAşımıYaşına ayarlanmalıdır. Aksi halde kalan yaşam süresi, PTSE’nin ilk yüklenen kalan yaşam süresinden geçen zaman çıkarılarak hesaplanır.

#### **4.3.4.2 Kendi Kendine Oluşan PTSE’lerin Yenilenmesi**

Kendi kendine başlayan bir PTSE’nin kalan yaşam süresinin, “Başlangıç Yaşam Süresi – PTSEYenilemeAralığı” farkının altına düşmesi, PTSE’nin en azından PTSEYenilemeAralığında yenilenmediğini gösterir, bu durumda PTSE yeniden başlatılır.

#### **4.3.5 Topoloji Bilgisinin Tetiklenmiş Güncellemeleri**

Değişmeyen ve yenilenmeyen PNNI topoloji bilgileri, yaşlanarak bağdan çıkmasını önlemek için periyodik olarak yeniden başlatılmalıdır.

Bir PNNI elemanının PTSE’lerinden birini yeniden oluşturacağı bir başka ek durum, önemli bir değişikliğe cevaptır. Herhangi bir bilgi grubunun (IG) içeriğinde meydana gelen önemli bir değişiklik, onu içeren PNNI Topoloji Durum Elemanı (PTSE) için de önemli bir değişiklik oluşturur. Önemli bir değişikliğe cevap olarak yapılan bu tür bir güncellemeye tetiklenen güncelleme denir.

PTSE’lerin ardışık yeniden oluşturulması arasındaki süre, MinPTSEAralığından kısa olmamalıdır. MinPTSEAralığı, temel olarak, bir düğümün ağa kabul edilemeyecek kadar yüksek seviyede yeni PTSE’leri göndermesini engelleyen bir bekletme sayacı olarak çalışır.

##### **4.3.5.1 PNNI Topoloji Durum Elemanları İçin Önemli Değişiklikler**

PTSE’ler, değişen sayıda IG’ler içerir. Bir PTSE’ye yeni bir IG’nin eklenmesi, PTSE için önemli bir değişikliğe sebep olur. PTSE’den bir IG’nin silinmesi de aynı şekilde önemli bir değişikliğe yol açar. PTSE’deki son IG’nin silinmesi, ele alınan PTSE’nin erken yaşlanmasıyla sonuçlanan önemli bir değişikliğe sebep olur. Son olarak, bir

PTSE içinde bulunan herhangi bir IG'deki değişiklik, o PTSE için önemli bir değişikliğe yol açar.

IG elemanlarının içeriğindeki önemli değişikliğin tanımı, IG'de bulunan bilginin türüne bağlıdır. Altı tür IG vardır:

- Düşüme ait bilgiler
- Dahili Ulaşılabilir ATM Adresleri
- Harici Ulaşılabilir ATM Adresleri
- Düşüme Ait Durum Parametreleri
- Yatay Bağlar
- Üst bağlar

Kaynak Kullanılabilirliği bilgi grubu (bazen opsiyoneldir) bir IG içinde bulunur. Kaynak Kullanılabilirliği bilgi grubunun içeriklerinde yapılan bir değişiklik, IG içeriğine yapılmış bir değişiklik olarak kabul edilir. Bu yüzden, bir RAIG'e yapılan herhangi bir değişiklik, kendisini içeren PTSE'de de önemli bir değişiklik yaratır.

#### **4.3.5.2 Önemli Değişikliklerin PTSE'ye İşlenmesi**

En son PTSE oluştuğundan beri, eğer PTSE MinPTSEAralığından daha uzun bir süre geçmişse, önemli bir değişiklik sonucu yeni bir PTSE oluşturulabilir. Eğer ele alınan PTSE MinPTSEAralık zamanından daha kısa süre önce oluşturulmuşsa, yeniden PTSE oluşturulmaz. Düşüm, PTSE'yi tekrar başlatmadan önce MinPTSEAralık zamanının geçmesini bekler. MinPTSEAralık zamanı, geçtiğinde bahsedilen PTSE tekrar başlatılır.

#### **Düşüme Ait Bilgi Gruplarındaki Değişikler**

Düşüme ait bilgi gruplarındaki herhangi bir değişiklik, önemli bir değişikliktir.

#### **Dahili Ulaşılabilir ATM Adreslerin IG'lerindeki Değişiklikler**

Dahili Ulaşılabilir ATM Adreslerin IG'lerindeki herhangi bir değişiklik önemli bir değişikliktir. Bu, dahili adreslerin ulaşılabilirken, ulaşılamaz (veya tam tersi) duruma

geçtiğinde görülür.

### **Harici Ulaşılabilir ATM Adreslerin IG'lerindeki Değişiklikler**

Harici Ulaşılabilir ATM Adreslerin IG'lerindeki herhangi bir değişiklik önemli bir değişikliktir. Bu, harici adreslerin ulaşılabilirken, ulaşılamaz (veya tam tersi) duruma geçtiğinde görülür.

### **Diğer Bilgi Gruplarındaki Değişikler**

Bu bölümde geri kalan IG'lerdeki değişiklikler anlatılmıştır. Aşağıdaki IG'lerden herhangi birinin eklenmesi veya silinmesi önemli değişiklik olarak kabul edilir:

- Düğüme Ait Durum Parametreleri IG'leri
- Yatay Bağ IG'leri
- Üst bağ IG'leri

Bu IG'lerden her biri Kaynak Kullanılabilirliği bilgi grubunu içerir. Kaynak Kullanılabilirliği bilgi grubundaki herhangi bir değişiklik, IG için önemli bir değişiklik olarak ele alınır.

Bu IG'ler düğüm, kapı ve peer grup belirteçleri içerirler. Bu alanlarda oluşan herhangi bir değişiklik önemli olarak düşünülür. Yatay Bağ ve Üst bağ IG'ler de Toplanma Göstergesi içerirler. Toplanma Göstergesindeki bir değişiklik, önemli bir değişikliktir.

Üst bağ Bilgi Özelliği (ULIA), sadece geri kalan içerik ULIA'nın yaratıcısı tarafından tarif edilen önemli bir değişikliğe uğradığında değişen bir sıra numarası içerir. Bu yüzden bir üst bağ ilanında, ULIA sıra numarasındaki bir değişiklik önemlidir, ULIA içeriğindeki başka değişiklikler önemsizdir.

### **Kaynak Kullanılabilirliği Bilgisindeki Değişiklikler**

Aşağıdaki bölümde, RAIG'nin hangi içeriklerindeki değişikliklerin önemli olduğu anlatılmaktadır[10].

### **Yönetimsel Ağırlık (AW)**

AW gerekli bir metriktir. Bu metriktaki herhangi bir değişiklik önemlidir.

### **Hücre Kaybı Oranı (CLRo)**

(CLRo) gerekli bir özelliktir. Herhangi bir değişiklik önemlidir.

### **Kullanılabilir hücre oranı (AvCR)**

Genel olarak, AvCR sıfıra yaklaştıkça, AvCR'deki değişiklikler daha önemli hale gelir. Örneğin, AvCR'nin sıfırdan, sıfırdan farklı değerlere geçişleri, eskiden taşınamayan bazı çağrılarının artık taşınabileceğini anlatır; bu ise kesinlikle önemli bir değişikliktir. Aynı şekilde, AvCR'nin sıfırdan farklı değerlerden, sıfıra geçişi, eskiden taşınabilen çağrılarının artık taşınamayacağı anlamına gelir ki bu da önemli bir değişikliktir.

AvCR'deki değişiklikler, ilan edilen son değere göre oluşan farklılıkla ölçülür. AvCR için önceki bir değer verildiğinde kullanılan algoritma, önemsiz durumları kapsayan bir üst bir de alt sınır kurar. Sınırlar içinde olduğu hesaplanan herhangi bir yeni değer, önemli bir değişiklik değildir. AvCR için, sınırların dışında olan tüm değerler, önemli değişikliklerdir.

### **Maksimum Hücre Aktarım Gecikmesi (maxCTD)**

MaxCTD da gerekli bir metriktir. MaxCTD'deki önemli bir değişikliği belirlemek için kullanılan algoritma, AvCR için kullanılanla çok benzer, sadece gecikme için bir üst sınır yoktur.

MaxCTD'deki değişiklik, ilan edilen en son değere, orantılı farklılık belirlenerek ölçülür. MaxCTD için önceki bir değer verildiğinde kullanılan algoritma, önemsiz durumları kapsayan bir üst bir de alt sınır belirler. Sınırlar içinde olduğu hesaplanan herhangi bir yeni değer, önemli bir değişiklik değildir. MaxCTD için, sınırların dışında olan değerler ise, önemli değişikliklerdir.

### **Hücre Gecikme Değişimi (CDV)**

CDV de gerekli bir metriktir. CDV için önemli bir değişikliği belirlemek için kullanılan algoritma, maxCTD için kullanılanla aynıdır.

CDV'deki değişiklik, ilan edilen en son değere, orantılı farklılık belirlenerek ölçülür. CDV için önceki bir değer verildiğinde algoritma, önemsizlikleri belirten değerler bir üst bir de alt sınır kurar. Sınırlar içinde olduğu hesaplanan CDV için herhangi bir

yeni deęer, önceki deęerden, önemli bir deęişiklik deęildir. CDV için Sınırların dışında olan deęerler, önemli deęişikliklerdir.

#### **Maksimum Hücre Oranı (maxCR)**

MaxCR gerekli bir özelliktir. MaxCR'deki herhangi bir deęişiklik önemlidir.

#### **Hücre Oranı Marjini (CRM) ve Deęişim Faktörü (VF)**

CRM ve VF ise isteęe baęlı özelliklerdir. CRM ve VF'deki deęişikliklere önemli gözüyle bakılmaz.

### **4.4 Hiyerarşı**

#### **4.4.1 Peer Grup Lideri**

PNNI hiyerarşisi, her peer grupta LGN'nin bazı işlevlerini uygulamak için bir düęümün seçilmesini gerektirir. Bu amaç için seçilmiş düęüme Peer Grup Lideri denir. Peer grup liderliği için tercih, konfigürasyonla kurulur. Bu tercih, her düęüm tarafından ilan edilen PGL öncelięi ile belirtilir. Düęüm Kimliği, aynı PGL öncelięine sahip düęümler arasında seçim yapmak için kullanılır.

##### **4.4.1.1 Peer Grup Lideri Seçim Algoritması**

PGL seçim algoritması, çıkan PGL'nin yerini almak için veya bir peer grup içinde peer grup liderliğini üstlenmek için uygun düęümü, dinamik olarak seçmede kullanılır. Düęüm, baęlantısı olduęu tüm düęümler arasında, PGL'si sıfır olmayan ve PGL'si en yüksek düęüme oy vermelidir. Liderlik öncelięi, peer grupta tüm düęümler tarafından PTSE'lerle ilan edilir. Hiçbir düęüm sıfırdan farklı PGL öncelięi ilan etmezse, hiçbir düęüm PGL olarak seçilmez. Bir düęüm, PGL seçimi için yeterli olan 2/3 oy üstünlüğünü, yeterli süre içinde ortak bir seçimin elde edilemeyeceęine karar verdikten sonra yeterli görecektir. Böylece, peer gruptaki az sayıda düęüm, yanlış uygulamalar, hatalı seçimlere sebep vermeyecektir. Aynı sebepten, peer gruptaki tüm düęümler normal işlem esnasında PGL seçimine katılmalıdırlar. Ama, Düęümsel Kimliğinde PGL Seçim bayraęı, geçiş yok'a ayarlı olan düęümler PGL seçimine katılmaz, PGL Seçim FSM'sini çalıştırmaz ve Liderlik Öncelięi ile Tercih Edilen PGL için sıfır ilan eder. Bu tür düęümler, peer grup içinde

bağlanabilirliğe karar verilirken de hesaba katılmazlar, bu tip aşırı yüklü düğümlerce ilan edilen PGL önceliği ve tercih edilen PGL, diğer düğümler tarafından yok sayılır.

PNNI protokolü, lider seçimi algoritmasını, kullanılan bağların güvenilir olduğu varsayımına göre geliştirilmiştir. Oysaki bağ hatalarının varlığı lider seçim algoritmalarını olumsuz etkiler. Bağ hatalarına karşı toleranslı lider seçim algoritmaları bulunmaktadır[11].

#### **4.4.1.2 PGL Seçim Veri Yapısı**

Her düğümün tek bir Peer Grup Lideri seçimi veri yapısı bulunur. Bu, aşağıdaki maddelerden oluşur:

**Durum :** Peer grup lideri seçim FSM'nin işlemsel durumu.

**TercihEdilenPeerGrupLideri :** Bu düğümün, peer grup lideri olmasını istediği düğümün Kimliği. Tercih ettiği peer grup liderini seçmek için düğüm, bu düğümden ulaşılabilir düğümler için topoloji veritabanında bulunan PTSE'lerdeki liderlik önceliklerini ve düğüm Kimliklerini kıyaslar.

**TercihEdilenPGLLiderliğiÖnceliği :** Tercih edilen peer grup liderinin liderlik önceliğidir.

**PeerArama Zamanlayıcısı :** Eylemsizlik Faktörü çarpı Hello Aralığını tetikleyen dahili zamanlayıcıdır. Bu zamanlayıcı tetiklendiğinde düğüm, peer'ı olmadığına karar verir ve derhal seçim işlemine başlar.

**PGLBaşlama Zamanlayıcısı:** PGLBaşlamaZamanını tetikleyen dahili zamanlayıcıdır. Bu zamanlayıcı tetiklendiğinde seçim süreci başlar. Düğüm bir TercihEdilenPeerGrupLideri seçer ve Düğüme Ait Bilgi PTSE'sinin yeni bir örneğinde seçimini ilan eder. PGLBaşlama Zamanlayıcısı, her düğümün, sadece topoloji bilgisinin tüm peer gruba yayılması için yeterli bir süre bekledikten sonra oyunu bildirmesini sağlar.

**Oy Birliği Zamanlayıcısı :** OyBirliğiGecikmesini tetikleyen dahili zamanlayıcıdır. Düğümlerin oy birliğini sonsuza dek beklemesini engeller.

YenidenSeçim Zamanlayıcısı : PGL ile bağlantı kaybedildiğinde başlayan ve YenidenSeçimAralığından sonra tetiklenen dahili zamanlayıcıdır. Bu zamandan sonra, seçim işlemi tekrar başlamalıdır. Düğüm yeni bir tercih edilen peer grup lideri seçer ve seçimini Düğüme Ait Bilgi PTSE'sinin yeni bir örneği içinde ilan eder. YenidenSeçim Zamanlayıcısı, düğümlerin PGL'ye olan bağlanabilirliğin kaybı durumunda, peer gruptaki düğümünlerin güncellenmiş topoloji bilgisini alma şansına sahip olana kadar, bir yeni PGL seçmesini geciktirir. Bu da, sistemin tutarlılığını önemli ölçüde artırır.

#### 4.4.1.3 PGL Seçim Durumları

Peer Grup Lideri seçim FSM'sinin ulaşabileceği durumlar bu bölümde anlatılmıştır. Şekil 4.3'de olası durum değişiklikleri verilmiştir. Yaylar, durum değişikliğine sebep olan olaylarla etiketlenmiştir.

Başlatılıyor : Durum makinesinin başlangıç hali.

Bekliyor : Düğüm, en azından bir bağda Hello FSM'ye başladı. Henüz bir peer bulunmadı.

Dolu Bekliyor : En azından bir komşu peer bulundu, henüz hiçbir veritabanı zamanuyumlaması tamamlanmadı.

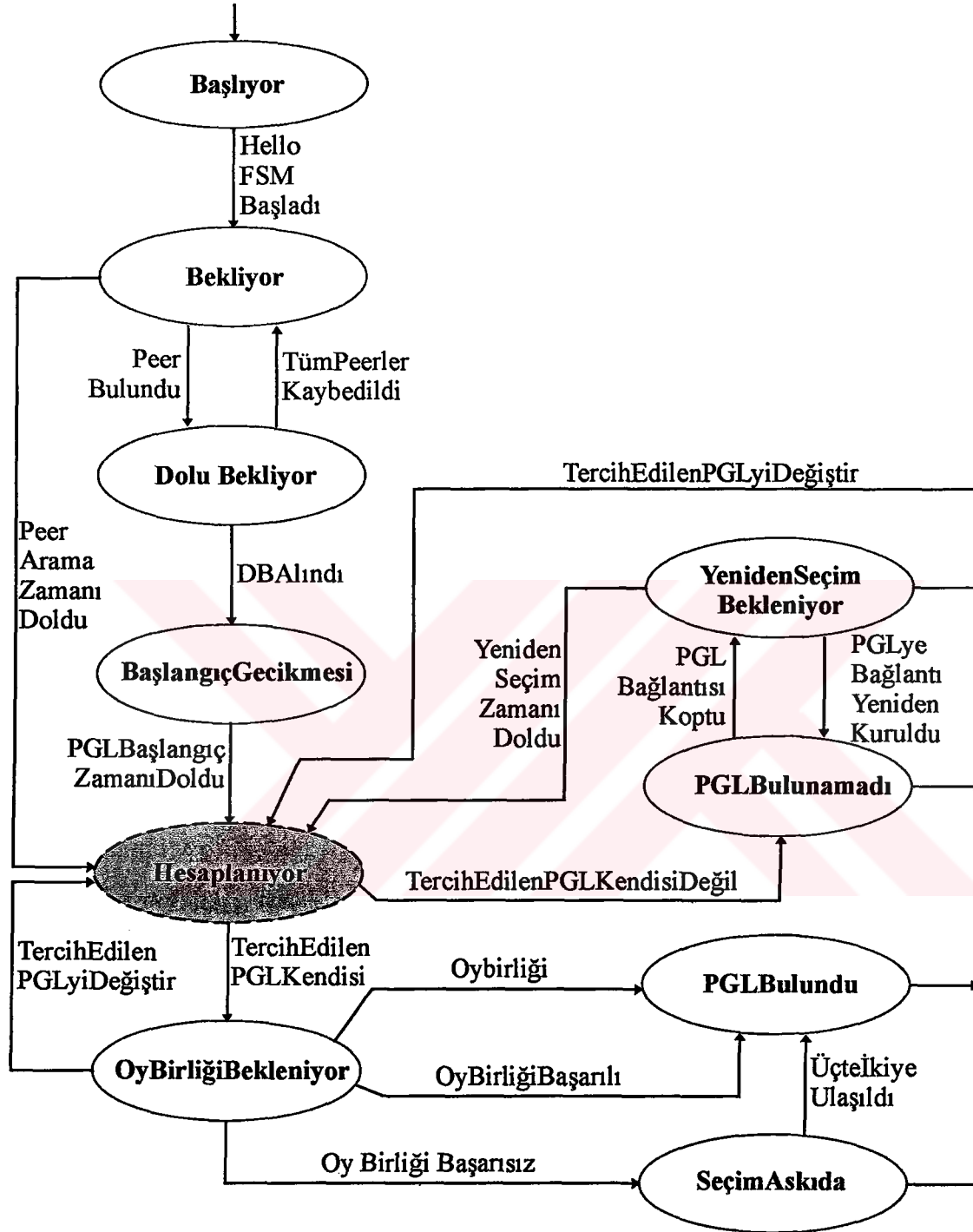
Başlangıç Gecikmesi : En azından bir komşu peer'la veritabanı zamanuyumlaması tamamlandı ve PGL Başlama Zamanlayıcısı başladı. Tercih ettiği PGL'yi seçip ilan etmeden önce, düğüm PGL Başlama zamanını beklemelidir.

Hesaplanıyor : Düğüm, TercihEdilenPGL için (yeni) seçimin ne olduğunu hesaplama sürecindedir. Bu durum geçici bir durumdur ve düğüm PGL için seçimini yapar yapmaz bu durumdan çıkılır. Bu seçim, aşağıdaki TercihEdilenPGLyiSeç algoritması uygulanarak yapılır.

Hesaplama durumdan diğer durumlara geçiş, düğümün kendisini ya da başka bir düğümü PGL olarak seçip seçmemesine bağlıdır. Geçiş, aşağıdaki iki durumdan biri olduğunda gerçekleşir:

- i) Başka bir düğüm seçilmişse, TercihEdilenPGLKendisiDeğil

ii) D ğ m kendini se mi se, TercihEdilenPGLKendisi



 ekil 4.3 PGL Se im FSM'i

**PGLBulunamadı :** Bu d ğ m, peer grup lideri de ildir. PGL olmak i in hangi d ğ m n en y ksek liderlik  nceli ine sahip oldu unu anlamak i in, di er d ğ mler tarafından g nderilen PTSE'leri incelemeye devam eder.

**PGLBulundu** : Bu düğüm, peer grup lideridir. Kendisinden daha yüksek liderlik önceliği olan düğüm var mı diye diğer düğümler tarafından gönderilen PTSE'leri incelemeye devam eder.

**OyBirliğiBekleniyor** : Düğüm kendisini Peer Grup Lideri olarak seçmiştir. Bu duruma girdiğinde, düğüm ilk olarak derhal oy birliği ile mi seçildiğini kontrol etmelidir, eğer öyle ise, bir Oy Birliği olayı oluşturmalıdır. Kendisini peer grup lideri olarak tanıtmadan önce oy birliğini veya Oy Birliği Zamanlayıcısının tükenmesini bekler.

**SeçimAskıda** : Düğüm kendisini peer grup lideri seçmiştir ama Oy Birliği Zamanlayıcısı tükendikten sonra diğer düğümlerin 2/3'ünden azı, onu tercih edilen PGL'leri olarak ilan etmiştir. Bu durum, topolojideki veya ağ parametrelerindeki değişikliklerden sonra meydana gelebilir. Bu durum kendi kendine düzelecektir, ya düğüm tercih edilen Peer Grup Lideri seçimini değiştirecektir ya da diğer düğümler bu düğümü PGL olarak kabul edeceklerdir.

**YenidenSeçimBekleniyor** : Düğüm o anki PGL ile olan bağlantısını kaybetmiştir. **YenidenSeçimZamanlayıcısı** başlamıştır. Zamanlayıcı tükenmeden önce bağlantı tekrar kurulmazsa PGL seçimi tekrar yapılır.

#### **4.4.1.4 Komşu Peer Durum Değişikliklerine Yol Açan Olaylar**

Durum değişikliklerine birçok olay sebep olabilir. Bu olaylar, Peer Grup Lider Seçimi ile ilgili prosedürlerle, özellikle bir düğüme ait bilgi grubu içeren PTSE'nin alınmasıyla gerçekleşir. Ayrıca, Hello durum makinelerinde ve Komşu Peer durum makinelerindeki gelişmelerle de tetiklenebilir. Olaylar, Şekil 4.3'de etiketli yaylar olarak gösterilmiştir.

**Hello FSM Başladı** : İlk Hello durum makinesi bağ üzerinde çalıştı.

**PeerBulundu** : Hello durum makinesi 2-Way Inside'a ulaştı.

**Tüm Peer'lar Kaybedildi** : 2- Way Inside durumundaki son Hello durum makinesi, bu durumdan çıktı.

PeerAramaZamanıDoldu : PeerArama Zamanlayıcısı tetiklendi. Hiçbir peer bulunamadı. Peer Grup Lider Seçimi derhal başladı.

DB Alındı : Bir Komşu Peer durum makinesi Dolu durumuna ulaştı.

PGLBaşlangıçZamanıDoldu : PGLBaşlama Zamanlayıcısı tetiklendi. Düğüm artık tercih ettiği peer grup liderini seçip, ilan edebilir.

TercihEdilenPGLKendisiDeğil : Hesaplanıyor durumuna giren düğüm, tercih ettiği PGL'nin kendisi olmadığına karar verir. Liderlik önceliği sıfıra eşitse veya peer'ın daha yüksek önceliği varsa bu durum gerçekleşir.

Oybirliği : Düğümün tercih ettiği PGL kendisidir ve peer grubundaki diğer tüm düğümler artık PTSE'lerinde, tercih ettikleri düğümün bu olduğunu belirtirler.

Oy Birliği Başarılı : Oy Birliği Başarılı zamanlayıcısı tetiklenir. Düğümün tercih ettiği PGL kendisidir. Peer gruptaki diğer düğümlerin 2/3 veya daha fazlası PTSE'lerinde, tercih ettikleri PGL'nin bu düğüm olduğunu belirtir.

Oy Birliği Başarısız : Oy Birliği Başarısız zamanlayıcısı tetiklenir. Düğümün tercih ettiği PGL kendisidir. Peer gruptaki diğer düğümlerin 2/3'den azı, PTSE'lerinde tercih ettikleri PGL'nin bu düğüm olduğunu belirtir.

Üçte İkiye Ulaşıldı : Düğümün tercih ettiği PGL kendisidir. Peer gruptaki diğer düğümlerin 2/3'ü PTSE'lerinde tercih ettikleri PGL'nin bu düğüm olduğunu belirtiyor.

TercihEdilenPGLKendisi : Hesaplanıyor durumuna giren düğüm PGL olması gerektiğine karar verir.

TercihEdilenPGLyiDeğiştir : Bu grubun tercih ettiği PGL'yi değiştiren yeni bilgiler alınır (örneğin, yeni bir yüksek öncelikli peer belirir; o anki PGL'nin önceliğini düşürür; veya yönetim bu peer'ın önceliğini, önceki peer'ın PGL'sinden yüksek bir değere yükseltir).

PGLBağlantısıKoptu : Peer Grup Lideri bağlantısı kaybedildi. Bir düğümün PGL'ye bağlanabilirliği olup olmadığına karar verilmesi (veya PGL seçimini başlatmak amacıyla başka bir düğüme) özel yönlendirme hesaplamalarını gerektirir. Yani, peer

gruptaki kaynak ve politika kısıtlarını yok sayarak, PGL'ye bağlanabilirlik var mı yok mu karar vermelidir. Her iki uçtaki düğümler tarafından ilan edilen tüm bağlar ve peer grupta ilan edilen tüm düğümler (düğümsel bayraklarında PGL Seçim bayrağı için geçişe kapalıya ayarlı olan düğümler dışında) PGL bağlanabilirliğini değerlendirmek için göz önüne alınmalıdır.

PGLyeBağlantıYenidenKuruldu : PGL ile bağlantı tekrar kuruldu.

YenidenSeçimZamanıDoldu : Yeniden Seçim Zamanlayıcısı tetiklendi. Bu durumda, seçim baştan yapılır.

#### **4.4.1.5 Dügüme Ait Bilgi PTSE'sinin Gönderilmesi**

Peer grup lider seçiminin başında düğüm, düğüme ait bilgi grubu içeren bir PTSE yaratmalıdır. Bu PTSE, yaşlanıp ağdan çıkmasını engellemek amacıyla, bir değişiklik meydana geldiğinde ya da düzenli olarak yeniden oluşturulmalıdır.

Bekliyor, Dolu Bekliyor ve Başlangıç Gecikmesi durumlarında ve düğüm PGL Seçim bayrağı için Non-Transit'e sahipse, tercih edilen peer grup liderinin Dügüm Kimlik alanı sıfıra ayarlanmalıdır. Dahası, düğüm PGL Seçim bayrağı için Non-Transit'e sahipse, Liderlik Önceliğini de sıfıra ayarlamalıdır. Hesaplanıyor, OyBirliği Bekleniyor, Yeniden Seçim Bekleniyor, PGLBulunamadı, PGLBulundu ve Seçim Askıda durumlarında, alan, veri TercihEdilenPeerGrupLideri'ne ayarlanmalıdır.

#### **4.4.1.6 Tercih edilen PGL'nin Seçilmesi**

Bir düğümün, düğüme ait bilgi grubunu içeren bir PTSE'yi her alışında ve kabul edişinde, TercihEdilenPeerGrupLideri değerini yeniden değerlendirmelidir. Bu düğüm, TercihEdilenPeerGrupLideri değerini, peer gruptaki başka bir düğüme bağlantı kuruluşunda ve kaybedilişinde de yeni baştan değerlendirmelidir.

Düğüm, peer gruptaki ulaşılabilir tüm düğümlerin liderlik önceliğini kıyaslar (kendisi de dahil). Veritabanında görünen ama bağlantıları olmayan düğümler (PGLBağlantısıKoptu tanımında anlatılan bağlanılabilirlik kurallarıyla) ve de peer grup liderlik önceliği sıfır olan düğümler, bu kıyaslamaya dahil edilmezler. En yüksek liderlik önceliğine sahip olan düğüm tercih edilen PGL olarak seçilir. Peer grupta liderlik önceliği sıfırdan farklı olan ulaşılabilir düğüm yoksa, tercih edilen

PGL Kimliği sıfıra ayarlanır. Önceliklerin eşit olması durumunda en son kararı almada, Düğüm Kimliği kullanılır. Bu amaçla, 22-sekizlik Düğüm Kimliğine, Düğüm Kimliğinin ilk sekizlisi en önemli sekizli olmak kaydıyla, işaretli bir tam sayı gözüyle bakılır. Eşitlik durumunda, en yüksek liderlik önceliğine sahip düğümlerin oluşturduğu kümenin içinden en yüksek düğüm Kimliği değerine sahip düğüm, tercih edilen PGL olarak seçilir.

Bir düğüm sadece, diğer düğümlerden direkt olarak alınan bilgileri dikkate almalıdır. Özellikle, diğer düğümler tarafından ilan edilen tercih edilen PGL, bu düğümün tercih edilen PGL seçimini etkilememelidir.

#### 4.4.1.7 Liderlik Önceliği

Peer grubu liderlik önceliği, 0 ile MaxLiderlik arasında bir değerdir. Bu değer düğümün PGL olması yönündeki göreceli tercihi belirtmek için atanmıştır.

Esnekliği ve tutarlılığı sağlamak için, düğüm peer grup gibi çalışacağına karar verdiğinde (örneğin, PGL Seçim FSM durum OperPGL'ye girip “ben liderim” bitini ayarladığında), ilan edilen Peer grubu liderlik önceliğini sabit GrupLiderArtışı ile artırır.

Peer grup olmak istemeyen veya olamayan bir düğüm için, öncelik değeri 0 rezerve edilmiştir. Liderlik önceliği 0 olarak ilan eden bir düğümü, hiçbir düğüm peer grup lideri olarak seçmez.

Hiçbir düğüm, MaxLiderlik eksi GrupLiderArtışından daha büyük bir değerle konfigüre edilmez. Aktif bir PGL, liderlik önceliğini, önceki konfigüre edilmiş öncelik değerine GrupLiderArtışını ekleyerek ilan eder.

Peer grup olması için zorlanacak bir düğüme sahip olmayan peer gruplarında, lider olabilecek düğümler için peer grup liderlik önceliği 1 ile GrupLiderArtışı-1 arasında atanmalıdır.

İlk sıradaki düğümler üzerinde belli düğümlerin öncelik alacağı durumlarda (bir tanesi zaten PGL olsa bile) öncelik alacak düğümlere, “2\*GrupLiderArtışı” ile “3\*GrupLiderArtışı-1” arasında bir liderlik önceliği değeri atanmalıdır.

Son olarak, katı bir efendi lider gerekli ise, bu düğümün liderlik önceliği 4\* Grup Lider Artışı ile (Maks Liderlik - GrupLiderArtışı) arasında bir değere atanmalıdır. Bu, ikinci lider sırasından öncelikli bir değer üretecektir ve hala arttırılabilir olacaktır.

Aynı anahtarlama sisteminde başlayan iki farklı hiyerarşi seviyesindeki iki düğümün peer grup liderlik öncelikleri arasında dahili bir ilişki yoktur. Bir anahtarlama sistemi, sadece bir peer grubunda düğüm olarak mevcutsa PGL olabilir. Bu, üye olduğu alt seviye gruplarının tümünün lideri olması gerektiği anlamına gelmez. Bu yüzden, bir düğüm hiyerarşinin bir seviyesinde PGL olarak önceden belirlenmişse, bu düğümün aynı anahtarlama sistemi içinde başlatılmış tüm ata düğümleri varlıklarını sona erdirirler. Varlıkları sona eren bu ata düğümlerin de biri PGL ise, bu durum, ilgili peer gruplarda yeni bir PGL'nin seçilmesini zorlar. Bir anahtar sistemin liderlik önceliği, hiyerarşinin her seviyesinde farklı olabilir.

Yukarıdaki özel durumdaki gibi, tüm peer grup liderlerinin üst seviye peer grup liderleri olmak istemesi gerekmez. Bir düğüm üst seviye peer grup kimliğini bilmiyorsa, lider olmamalıdır. Ayrıca, yerel sebeplerden ötürü, üst seviye lideri olmamak da isteyebilir. Her iki durumda da, o hiyerarşi seviyesinde düğüm, liderlik önceliği olarak 0 ilan eder.

#### **4.4.1.8 Dış Bağlarda Düğüme Ait Hiyerarşi Bilgilerinin Değişimi**

Peer grup lideri seçildiğinde ve bunu “Ben Liderim” bitini 1 olarak ayarlayıp Düğüme Ait Bilgi PTSE'sinde ilan ettiğinde sınır düğümler, dış komşularla değiş-tokuş edilen Hellolardaki düğümsel hiyerarşi listesindeki PGL'den alınan Üst Seviye Birleştirme bilgilerini içerirler.

Eğer birkaç düğüm bu peer grubun lideri olmayı isterse, sınır düğümler, yerel olarak peer grup lideri seçilenin bilgilerini ilan eder. Üst-seviye peer grupları için, eğer birden fazla düğüm PGL olmayı isterse, sınır düğümler, o peer grup içinde ata LGN tarafından seçilenden aldığı bilgiyi ilan eder.

#### **4.4.2 Üst Bağların İlanı**

Bir üst bağ ilanı oluşturulduğunda ve peer gruba sokulduğunda, her iki yönde de topoloji durum parametrelerini içermelidir, böylece güzergah seçimi sırasında PNNI, bu bağları dikkate alabilir. Bu topoloji parametreleri aşağıdaki şekilde elde edilir.

##### **4.4.2.1 En Düşük Seviye Düğümlerde Üst Bağların İlanı**

Bir dış bağın düğümleri, diğer düğümün başka bir peer gruba ait olduğunu fark ettiğinde, Hello paketlerin içine sınır dışı yönünde Kaynak Kullanılabilirliği bilgilerini de dahil etmelidirler. Bu bilgiyi taşıması için Hello Paketine bir ULIA eklenir.

Burada değiştirilecek topoloji durum parametreleri, bu bağ, peer grubun içinde olsaydı, ortaya çıkacak ilandaki topoloji durum parametreleri ile aynı olurdu. Tek fark, PTSE'ler yerine Hello paketlerinde taşınmalarıdır.

Hellolar, asla birden fazla bağda yayılmadıklarından topoloji durum parametreleri, Hellonun alındığı kapıyla ilgili dış bağa, alıcı tarafından bağlanabilirler. Başka bir birleştirme bilgisine gerek duyulmaz. O kapı üzerinde Hellolardan alınan topoloji durum parametreleri, bu dış bağın sınır içi yönündedir.

Hiyerarşi oluşumu tamamlandıktan sonra, dış bağlar üst bağların temeli oluştururlar. Üst bağlar, dış bağların her iki ucundaki sınır düğümlerinin peer gruplarına ilan edilirler. Sınır düğümleri, dış bağlardaki sınır içi yönleri için topoloji durum parametrelerini, dış bağların sınır içi yönü için topoloji durum parametrelerini hesaplamakta kullanırlar. Böylece, üst bağ ilanını bir peer gruba sokan sınır düğümü, her iki yön için de topoloji durum parametrelerini belirtir.

Bireysel üst bağlar, her dış bağ için oluşturulabilirler. Bu durumda, bir Hello içinde alınan topoloji durum parametreleri, ilgili üst bağ için değişmeden ilan edilir. Alternatif olarak, tek bir üst bağ, birkaç dış bağın toplanması ile oluşmuş da olabilir. Bu tip bir toplanma için prosedürler, genelde bağların toplanmasında kullanılan ile aynıdır.

#### 4.4.2.2 Üst Bağlarda Ters Yön Bilgisi

Üst bağ ilanları, dış bağlardan veya diğer üst bağlardan türer. Dış bağlardan türeyen üst bağlar için ters yön bilgileri (topoloji durum parametreleri de dahil), dış komşu düğüm tarafından belirtilir. Sınır düğümleri, dış komşunun üst bağ ilanlarında ters yön bilgisi olarak içereceği bilgileri, dış bağ üzerinden göndereceği Hellolara bir Üst bağ Bilgi Özelliği (ULIA) ekleyerek belirtir. Hellolardaki ULIA, alıcı dış komşunun yarattığı, üst bağ ilanında ilan edilecek tüm bilgileri kapsar.

Üst bağlar diğer üst bağlardan da türetilir. Bu durum, bir çocuk peer grubunda bir veya daha fazla sınır düğümünün LGN'nin peer grubundan daha üst seviyedeki bir üst düğüme bir üst bağ ilan ettiğinde meydana gelir. Bu durumda LGN, aynı üst düğüme bir sonuç üst bağ ilan eder.

Üst bağın tek bir dış üst bağdan türediği veya harekete geçirilmiş üst bağın aynı üst seviye düğümünde bulunduğu durumlarda, bu çoklu bağlar bu düğümden çıkan bir genel üst bağda toplanabilirler. Bu durumda, toplanan üst bağ için yaratılan ULIA'da bulunan bilgi grupları, dış bağlardan alınan veya harekete geçen üst bağlarda ilan edilen Helloların içindeki ULIA'larda bulunan bireysel bilgi gruplarından türetilirler. ULIA'ların içinde, toplanmakta olan herhangi bir bilinmeyen bilgi grubu, sonuç üst bağı için oluşturulan ilana dahil edilmişse, bu bilgi grubu ile ilişkili bilgi grup ekleri tarafından kontrol edilir. Bu düğüm, toplanan bağ için ilan edilen topoloji durum parametrelerini ürettiğinden, ürettiği değerde önemli bir değişiklik olduğunda bunu belirtmek zorundadır. Yani, toplanan üst bağ için uygun ULIA sıra numarası değerleri üretmelidir.

Düğümler sadece, eğer bağın tüm mecburi bilgi gruplarını anlıyorsa, yön hesaplamalarında üst bağları kullanabilir ve DTL'lerinde belirtebilirler. Buna ek olarak, eklerin kapsamı hakkındaki normal kurallara bir istisna olarak, düğüm üst bağın ULIA'sının içindeki tüm mecburi bilgi gruplarını da anlamalıdır. Örneğin, ULIA paketleyicisi içinde tanınmamış ve mecburi olarak ek verilmemiş bir bilgi grubu varsa, ilgili bağ, yön hesaplamaları tarafından kullanılmamalıdır.

#### 4.4.3 Topoloji Bilgilerinin Birleştirilmesi

Topoloji toplanması, baba peer grupta ilan edilen bilgi hacmini azaltmak için çocuk peer grubun topoloji bilgisinin özetlenmesi işlemidir. PNNI'nın büyük ağlara ölçeklenebilmesi için bilgilerin etkin olarak birleştirilmesi gereklidir.

Bağların birleştirilmesi, birçok paralel bağı tek bir üst seviye bağ olarak gösterilmesi işlemidir. Bu tür bir bağ için topoloji durum parametreleri, bireysel bağların topoloji durum parametrelerinin birleşiminden türer.

PNNI kompleks düğüm temsili, bir mantıksal düğüm içindeki bağlanabilirliği anlatmakta kullanılan esnek bir yapıdır. Bir mantıksal grup düğümü kompleks bir düğüm temsili ürettiğinde, oluşturduğu bu temsilin doğruluğu ve boyutu arasında bir seçim yapmak zorunda kalır. Bu duruma alternatif olarak, basit düğüm temsili kullanılabilir, bunda ise tüm LGN'ye kaynak kısıtlaması olmayan bir nokta gözüyle bakılır.

Toplanma topoloji tanımını türetmek için kullanılan algoritmalar, uyarlamaya özgüdür. Dolayısıyla buradaki seçim kararı üretici firmalara bırakılmıştır.

##### 4.4.3.1 Bağların Birleştirilmesi

Aynı iki peer grup arasındaki dış bağların birleştirilmesi, ele alınan dış bağların bittiği sınır düğümlerdeki Toplanma Göstergesi değeriyle konfigüre edilir. Bilgi birleştirilmesi hakkındaki bilgi, peer grup değiştiğinde tutarlılığın devamını sağlamak için, mantıksal grup düğümleri yerine, sınır düğümleri içine konfigüre edilir. Aynı Toplanma Gösterge değerine sahip bir çift mantıksal düğüm arasındaki tüm bağlar, LGN tarafından bir mantıksal düğüm olarak tanıtılmalıdır. Toplanma Göstergesinin kapsamı, bir çift mantıksal grup düğümü arasındaki bağlarla sınırlandırılmıştır. Aynı gösterge değeri, karışıklık olmadan, diğer çift düğümler arasında bağlar için kullanılır.

Toplanma göstergesi, bir dış bağ ile bağlanan iki en düşük seviye komşu düğümleri arasında Hello protokolünde haberleşir. İki en düşük seviye komşu düğümü arasındaki gösterge değeri, yeniden konfigüre edilmedikçe değişmez. Bu değere Konfigüre Edilmiş Toplanma Göstergesi de denir.

Konfigüre Edilmiş Toplanma Göstergesi karşılıklı olarak değiştirildiğinde her en düşük seviye komşu düğümü, Türetilen Bağ Toplanma Göstergesinin genel değerlerini belirlemek için aynı bilgidaki aynı algoritmayı kullanır, bu Türetilen Bağ Toplanma Göstergesi, ilgili üst bağlar için PTSE’lerde ilan edilen değerdir. En düşük seviye düğümler arasındaki Hellolar Konfigüre Edilmiş Toplanma Göstergesinin değerini içerir; PTSE’ler ve mantıksal grup düğümleri arasındaki Hellolar her zaman Türetilen Bağ Toplanma Göstergesi değerini içerir.

Varsayılan değer olarak bağ, konfigüre toplanma göstergesini sıfıra ayarlar, bunun özel bir anlamı vardır. Herhangi başka bir değer ancak konfigürasyon esnasında kurulur. Özel bir değer olan sıfır değeri, bağın toplanma davranışını önemsemediğini gösterir. Bu durumda düğüm, uzak uç hangi değere sahip olursa olsun Türetilen Bağ Toplanma Göstergesi değerini kullanır. Hiçbir konfigürasyon yapılmamışsa, tüm bağlar, Türetilen Bağ Toplanma Göstergesi sıfır değeri ile biter.

Türetilen Bağ Toplanma Göstergesi hesaplamak için kullanılan algoritma:

1. Eğer komşular aynı Konfigüre Edilmiş Toplanma Göstergesi değerlerini değiştirirse, bu değer Türetilen Bağ Toplanma Göstergesi değeridir.
2. Eğer komşunun Konfigüre Edilmiş Toplanma Göstergesinin değeri sıfırsa ve diğer komşunun Konfigüre Edilmiş Toplanma Göstergesi değeri sıfır değilse, sıfır olmayan değer Türetilen Bağ Toplanma Göstergesi değeridir.
3. Eğer komşuların Konfigüre Edilmiş Toplanma Göstergesi değerleri farklıysa ve her ikisi de sıfır değilse, Türetilen Bağ Toplanma Göstergesi değeri sıfırdır.

Yukarıdaki algoritma sürekli olarak çalışır. Konfigüre toplanma göstergesi değerindeki değişiklik etkisini göstermeye başladığında yeni değer derhal dış Hellolarda en düşük seviye komşu düğümle haberleşir. Eğer bu Türetilen Bağ Toplanma Göstergesinde bir değişikliğe yol açarsa, Türetilen Bağ Toplanma Göstergesi değerindeki değişikliğinden etkilenen bağları içeren tüm PTSE’lerin yeni örnekleri ürer. Bu olay, hiyerarşinin tamamında ani bir değişikliğe yol açabilir.

En düşük seviye düğümler için, aynı Türetilen Bağ Toplanma Göstergesi değerine sahip olan, aynı üst seviye komşu düğümüne üst bağlar, bu düğüm tarafından anlatılan yolla toplanabilir veya dağıtılabilirler. Ama, farklı Türetilen Bağ Toplanma

Göstergesi değerine sahip üst bağlar toplanmamalıdır. Eğer en düşük seviye düğümün, aynı seviyede olduğu mantıksal düğüm olan bir komşusu varsa, bu mantıksal düğüme, aynı Türetilen Bağ Toplanma Göstergesi değerli tüm yatay bağlar toplanır ve tek bir bağ olarak ilan edilirler.

Mantıksal grup düğümleri için, aynı Türetilen Bağ Toplanma Göstergesi değerine sahip olan düğüme tüm bağlar toplanmalı ve tek bir bağ olarak ilan edilmelidir. Yani, eğer bir peer grubun, aynı Türetilen Bağ Toplanma Göstergesi değeri olan düğüme birden fazla üst bağı varsa, peer grubu temsil eden mantıksal düğüm, bu üst bağları tek bir üst bağda veya yatay bağda toplamalıdır.

Konfigüre toplanma göstergesinin varsayılan değeri sıfır olduğundan, varsayılan davranış, iki mantıksal düğüm arasındaki tüm bağları, Türetilen Bağ Toplanma Göstergesi değeri sıfır olan tek bir bağda toplamaktır.

Uzak düğüm kimliği ile beraber Türetilen Bağ Toplanma Göstergesi değeri, hiyerarşi seviyeleri arasında birleştirici bilgi görevi görür. Bu değerler, üst bağları veya yatay bağları anlatan PTSE'ler vasıtasıyla aktarılırlar. Üst bağ PTSE'leri aracılığıyla ilan edilen bu değerler, bu düğümün baba peer'ı tarafından sonraki hiyerarşi seviyesinde daha fazla toplanma kararları vermek için kullanılır. Buna ek olarak, üst bağlarda veya yatay bağlarda ilan edilen değerler DTL işlemi sırasında sınır düğümleri tarafından kullanılır. Bir giriş sınır düğümü, en üst DTL'nin o anki girişindeki sıfır olmayan kapı kimliğine sahip olan alınan bir DTL yığını işlerken, bu peer grubundan seçilen güzergah bu kapı Kimliği tarafından belirtilen bağ içine toplanan bir üst bağda son bulmalıdır (örneğin, peer grubundan çıkan üst bağ, DTL'de belirtilen kapı Kimliği değeriyle aynı toplanma göstergesi değerine sahiptir).

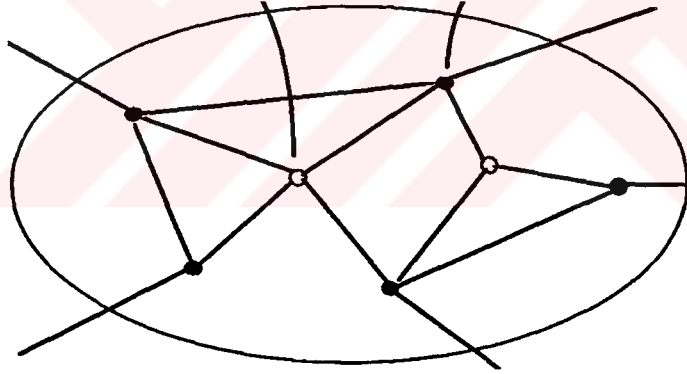
Formatın tutarlılığının korunması için, türemiş toplanma göstergesi alanı, yatay bağ PTSE'leriyle, fiziksel bağlar ve VPC'ler için de taşınır. Bu gösterge, gönderilmeden önce sıfıra ayarlanmalıdır. Bu yatay bağ altında harekete geçirilmiş üst bağlar bulunmadığından, bu yatay bağ için ilan edilen toplanma göstergesinin değeri etkisizdir.

#### 4.4.3.2 Basit D ğ m Temsili

Bir baęlantının noktadan-noktaya QoS ve trafik parametre deęerleri, bu baęlantının getięi t m d ğ mlerden ve baęlardan etkilenir. Bir d ğ mden geiřin, baęlantının noktadan-noktaya QoS ve trafik parametre deęerlerini  nemsenmeyecek kadar az olarak etkiledięi durumlarda, en basit yaklařım, bu t rde bir d ğ m n tek bir nokta olarak řekillendirilmesidir. Bu temsil sayesinde, y nlendirme hesapları nisbeten daha kısa s rede yapılır; temsilde aę topolojisi k řelerle, baęlar yaylarla g sterilir. Bu temsil  zellikle tek anahtarla uygulanan en d ř k seviye d ğ mleri řekillendirmek iin kullanıřlıdır, burada d ğ me baęlı baęların sınırlamaları, anahtarın kendi sınırlamalarından ok daha  nemlidir.

#### 4.4.3.3 Kompleks D ğ m Temsili

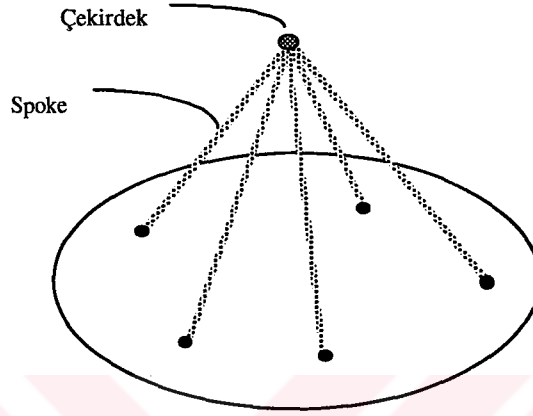
Kompleks d ğ m temsili, bir mantıksal d ğ mle ilgili detaylı durum bilgileri saęlayan d ğ me ait durum parametreleridir. Bir d ğ m n iinden veya iine doęru geiř yapıldıęı durumda d ğ m n, baęlantının noktadan-noktaya parametre deęerleri  zerinde  nemli bir etkisi vardır.



Şekil 4.4 Beř sınır d ğ m ne sahip bir peer grup

Bir mantıksal d ğ mden geiřin y nlendirilmesinin sanki bir d ğ mden geermiř gibi olmasını saęlamak iin,  niform yarıaplı simetrik yıldız topolojisi kullanılır. Yıldızın merkezi, mantıksal d ğ m n dahili referans noktasıdır ve buna ekirdek adı verilir. ekirdek ile mantıksal d ğ m n bir kapısı arasındaki mantıksal baęlantıya ise spoke adı verilir. PNNI Y nlendirmesi, her d ğ msel durum parametresi iin her iki y nde de mantıksal d ğ m n ıkıřı ve giriři arasında tahmini tek bir deęere sahip olan, varsayılan bir d ğ m temsilini destekler (Şekil 4.5).

Mantıksal düğüm ile ilgili tüm düğümsel durum parametreleri için, bir “yarıçap” mantıksal düğümün “çapından” türetilir. Düğümsel bir özellik için, “yarıçap”, “çap” ile aynıdır. PNNI yönlendirmesi, “çapın” belirlenmesi amacıyla toplama işleminin nasıl yapılacağını tam olarak belirtmez. Bu durumda ya en kötü durum değerleri ilan edilerek tutucu bir politika ya da ortalama durumu, hatta en iyi durumu yansıtan değerler kullanılarak da agresif bir politika izlenebilir.



Şekil 4.5 Varsayılan Düğüm Temsili

Varsayılan düğüm temsili, ilan edilen bilgileri büyük ölçüde indirgeme imkanı sunar. Fakat tipik mantıksal düğümdeki çoklu bağlanabilirliği yakalayamaz ve asimetrik topoloji bilgilerini tam olarak yansıtamaz (Şekil 4.5).

Bir mantıksal düğüm tamamen yuvarlak olmadığından, PNNI Yönlendirmesi, herhangi bir spoke ile ilgili topoloji durum parametrelerinin, varsayılan “yarıçaptan” farklı olmasına izin verir. Buna ek olarak, bypass adı verilen, direkt kapıdan-kapıya bağlantılar da ilan edilebilir.

Varsayılan düğüm temsili dışında bir şey temsil eden bağlanabilirlik ilanına, istisna adı verilir (Şekil 4.6).

PNNI Yönlendirmesi için kompleks düğüm temsili aşağıda anlatıldığı şekilde kurulabilir:

1. Kavramsal olarak, her mantıksal düğüm üzerinde, bu düğümün içini temsil eden bir çekirdeğe ve bu mantıksal düğümün kapılarını, çekirdekle bağlayan spoke'lara sahip, bir yıldız topolojisi yatar.



#### 4.5 Güzergah Seçimi

ATM, bağlantılı yönlendirmeyi destekleyen bir ağ teknolojisidir. Bunun anlamı, bir sanal bağlantı veya sanal güzergahın kurulması için PNNI tarafından seçilen güzergahın uzun bir süre kullanımda kalacağıdır. Yani, yetersiz bir yönlendirme kararının sonuçları, açık kaldıkça bağlantıyı etkileyecektir. Bu yüzden PNNI'nın güzergahları dikkatle seçmesi önemlidir.

ATM bir bağlantı kurulurken kullanıcıya çağrı aşamasında hizmet kalitesi (QoS) ve bant genişliği parametrelerini belirtmesine izin verir. ATM ağı o çağrı için istenen hizmet kalitesi (QoS) ve bant genişliği parametrelerini garanti edebilmelidir. Çağrı kuruluşu iki aşamadan oluşur: (1) güzergah seçimi, ve (2) o güzergah boyunca her noktanın bağlantı durumunun ayarlanması. Güzergah seçimi öyle bir şekilde yapılır ki, seçilmiş güzergah, o an kullanılabilen bilgilere bağlı olarak, gereken QoS ve bant genişliğini destekleyecek kapasitede görünür. Güzergah boyunca her düğümün çağrı kuruluşunun işlenmesi, gereken kaynakların aslında mevcut olduğunu onaylar. (Mevcut değilse crankback meydana gelir ve mümkünse yeni bir güzergahın hesaplanmasına yol açar; böylece en son sonuç ya ihtiyacı karşılayan bir güzergahın kurulması ya da çağrının geri çevrilmesi olur).

Etkin QoS-duyarlı güzergah seçimi hala araştırma konusudur. Ayrıca, birden fazla kısıtın varolduğu QoS-duyarlı güzergah seçimi için kullanılan bazı algoritmalar, birden fazla bağımsız bağ parametrelerinin ele alınmasını gerektirir ve gerektirdikleri işlem süresi açısından pahalıdır. Bu yüzden PNNI'nın QoS-duyarlı güzergah belirleme algoritmalarının seçiminde esnekliğe izin vermesi çok önemlidir.

PNNI yönlendirme protokolü bağ durumu yönlendirme algoritması temel alınarak geliştirilmiştir. Bu algoritmanın geniş çaplı ağları desteklemekteki dezavantajı hiyerarşik yönlendirme mekanizmaları eklenerek aşılmıştır. Yönlendirme için gerekli ulaşılabilirlik ve QoS bilgisi PNNI topoloji durum elemanları (PTSE) halinde ağ içine dağıtılır. Bu yapıyla OSPF gibi bağ durumu yönlendirme protokollerine benzerdir[13]. PNNI protokolü, düğümler tarafından iletilen büyük miktarlardaki bağ ve düğüm durum parametrelerini de destekler ayrıca QoS için kapsamlı bir desteğe sahiptir.

Ağ işlemlerinde kullanılan iki temel yönlendirme tekniği vardır: kaynaktan yönlendirme ve adım adım yönlendirme. Kaynaktan yönlendirmede, kaynak sistem hedefe giden güzergahı seçer ve güzergah üzerindeki diğer sistemler kaynağın belirlediği yönlendirme talimatlarına uyarlar. Adım adım yönlendirmede ise, her sistem o güzergah için bir sonraki adımı bağımsız olarak seçer ve her adımda verilen yönlendirme kararlarının tutarlı olması şartıyla, hedefe doğru ilerleme ile sonuçlanır.

Prensipte, her iki yönlendirme tekniği de ATM gibi bağlantı-temelli ağlara uygulanabilir. Ama bu türde bir ağda adım adım yönlendirmenin bazı dezavantajları vardır.

İlki yönlendirme güzergahında döngülerin oluşabilmesidir. Bu döngüler bir çok sebeple oluşabilir:

1. Anahtarların farklı yönlendirme algoritmaları kullanması durumunda alınacak yönlendirme kararlarındaki tutarsızlık,
2. Anahtarların sahip olduğu yönlendirme bilgilerini bulunduran veritabanları arasındaki bilgilerin tutarsızlığı (tipik olarak, ağdaki bir değişimin ağa henüz tam olarak yayılmamış, topoloji veri tabanlarının güncellenememiş olması)

Yönlendirme kararlarındaki tutarsızlık, adım adım yönlendirmede temel bir kısıtlamaya yolaçar; güzergah seçimi tamamen belirtilmelidir ve tüm sistemler belirtileni tam olarak uygulamalıdır.

Buna ek olarak, veri tabanları arasındaki tutarsızlık nedeniyle güzergahların, optimalden uzak ama döngü içermeyen, bireysel atlama kararlarının sonucu olarak oluşturulması da mümkündür. Bu tür bir yetersiz güzergahla kurulmuş herhangi bir bağlantı, bağlantı açık olduğu sürece bu güzergahı kullanacaktır.

Adım adım yönlendirmenin başka bir dezavantajı da, güzergah üzerindeki her sistemde güzergah seçimi işleminin maliyetinin tekrarlamaıdır. Bu bağlantı temelli ağlar için çok önemli olmasa da (çünkü maliyet sadece bir defa bağlantı kuruluşu sırasında ortaya çıkar), PNNI'in kullandığı QoS-duyarlı güzergah seçim algoritması çok daha pahalıya mal olabilir.

Kaynaktan yönlendirmede hedefe gidecek güzergahın seçiminden kaynak sorumludur. Bunu, ağ topolojisi hakkındaki yerel bilgiye dayanarak yapar. Sadece veritabanı bununla ilgili olduğundan döngülerin oluşması mümkün değildir. Dahası, güzergah sadece kaynak tarafından seçildiğinden kullanılan algoritmanın her sistem için aynı olması gerekmez ve bunu yapmak için gereken gider, sadece bir kez ortaya çıkar.

Bağlantı tabanlı hizmetlere bağlı olarak yukarıda anlatılan adım adım yönlendirmenin zorluklarıyla karşılaşmamak için PNNI tüm bağlantı kuruluş gereksinimlerinde kaynaktan yönlendirme kullanır. Yani, peer gruptaki ilk düğüm o peer gruptaki güzergahın tamamını seçer. Güzergah DTL olarak kodlanır, bu bağlantı kurma isteğine dahil edilir. DTL, güzergah üzerine kullanılan her düğümü ve isteğe bağlı olarak bu düğümler arasında kullanılan mantıksal bağları belirtir. Güzergah boyunca bir düğüm DTL'yi öz kaynak yetersizliği sebebiyle belirli bir bağlantı kuruluş talebi için kullanamıyorsa, düğüm bu talebi geri çevirmeli ve talebi DTL'yi yaratan düğüme crankback etmelidir.

PNNI birden fazla seviyeli hiyerarşik yönlendirmeye izin verdiğinden kaynak anahtar, belirlediği güzergaha bildiği tüm hiyerarşi detaylarını ekler. Buna hiyerarşik olarak tamamlanmış kaynak yön denir. Bu tür bir güzergah tamamen detaylı bir güzergah değildir çünkü oluşturana ait peer grubun dışındaki güzergahın detaylarını içermez. Bunun yerine güzergahın bu kısımları ardışık grup olarak özetlenir, anahtar sistemi o peer grubun transit geçişini anlatan bir (düşük seviye) kaynak rotayı seçmekten sorumludur. Doğal olarak, bir düşük seviye peer grubu geçmek için kullanılan güzergah, üst seviye güzergahı ile tutarlı olmalıdır (örneğin, üst seviye güzergah tarafından belirlenen “diğer durak” hedefine ulaşmalıdır).

Kaynaktan yönlendirmenin kullanılması, kurulacak bağlantı için güzergah seçiminde sadece bir düğümün karar vermesi anlamına gelir. Bu da bir peer gruptaki her düğümün aynı güzergah belirleme algoritmasını kullanma zorunluluğunu ortadan kaldırır. Farklı düğümler farklı güzergah seçim algoritmalarını birçok sebebe bağlı olarak kullanabilirler; örneğin düğümlerinin göreceli olarak farklı işlem kapasitesine sahip olduğu ağlarda, birçok marka ürün kullanılarak oluşturulmuş ağlarda veya yeni sürüm bir yazılıma geçişte. Benzer şekilde PNNI hiyerarşisinin bir seviyesinde

kullanılan güzergah seçim algoritması, başka bir seviye için kullanılandan farklı olabilir.

PNNI yönlendirmede kullanılan algoritmalar, örneğin Dijkstra algoritması gibi tek bir birikimli metrik kullanırlar. PNNI birçok kompleks bağ parametresi kullandığı için, güzergah seçimi sıkça kabul edilebilir bir güzergah oluşturamaz. Bu gibi durumlarda, gelişmiş algoritmalar kullanılabilir. Bu algoritmalar, bazı özellikleri (örneğin gecikme gibi) seçerek gevşeten ve güzergahları arzu edilen minimal özellikler kümesini karşılayacak şekilde yeniden hesaplayan algoritmalarlardır.

Tüm bu sebeplerden ötürü PNNI güzergah seçimi için kullanılacak tek bir algoritma belirtmez. Bunun yerine uygun görülen bir algoritmasının özgürce uygulanabilmesini sağlayan bir yapı sunar.

#### **4.5.1 Genel Bağlantı Kabul Denetimi**

Genel bağlantı kabul denetimi (GCAC), şu fikirden ortaya çıkmıştır: CAC standartlaştırılamayacağından bir anahtar sistemin yeni bir bağlantıyı kabul edip etmeyeceğini belirlemek için tek bir CAC uygulamak mümkün değildir.

Her bir bağlantı isteği için güzergah, bağlantıların hizmet kategorisi, trafik karakteristiği ve QoS gereksinimleri ile birlikte düğümün topoloji veritabanında depolanan bilgi baz alınarak hesaplanır. Güzergah, bağlantının uçtan uca ve QoS gereksinimlerini ve bunun yanında ağın etkinlik kriterini karşılayacak şekilde seçilir. Bağlantı kuruluşu sırasında seçilen güzergah boyunca yer alan her anahtar sistem, QoS garantilerini tehlikeye atmadan bağlantının kurulumunu sağlamak için kendi CAC'ını uygular, bu şekilde daha önceden bu düğüm üzerinde varolan bağlantılara sunulan hizmetlerin korunması garantilenir. Bir anahtar sistemin CAC'yi nasıl uyguladığı, standartlaştırma konusunun kapsamı dışındadır. Eğer bir anahtar sistemi bir bağlantı kabul etmişse, gelecekteki bağlantı taleplerini aynı şekilde kabul edip etmeyeceği değişebilir. Bu olay, oluşan değişimlerin önemli olması durumunda, düğümün güncelleştirilmiş yeni öz kaynak kullanımını anlatan yeni PTSE paketi üretmesini sağlar. “Önemli” bir değişimin belirlenmesi, konfigürasyon parametreleri tarafından kontrol edildiğinden algoritmik olarak tayin edilir. Konfigürasyon parametrelerinin değerleri, eski bilgiler nedeniyle başarısız olan çağrı denemesine karşı PTSE güncelleştirmelerinin sıklığını dengeleyecek bir şekilde belirlenir.

Bir genel CAC'a güzergah seçiminde ihtiyaç duyulmasının nedeni bir bağ veya bir düğümünün önerilen bağlantıyı destekleyecek yeterli öz kaynağının olup olmadığını anlamaktır. Bir genel CAC, sadece önerilen bağlantıyı kabul etme ihtimali yüksek düğümleri veya bağları güzergah listesine koyar. Önerilen bağlantıyı reddetme ihtimali yüksek düğüm veya bağları güzergah listesine konulmasını engeller. Aslında, bir genel CAC, bir anahtar sisteminde uygulanan o anki yerel CAC'nin sonuçlarını önceden tahmin etmede kullanılır.

Genel CAC kullanımı sayesinde her anahtar sistem istediği bir CAC kullanmakta serbesttir ama bunun karşılığı olarak genel CAC'nin, tahmini gerçekleştirebilmesi için gerekli bilgileri taşıyan bir topoloji durum paketini ağa düzenli olarak yayması gerekir. Kullanılan genel CAC ve onu destekleyen topoloji durum parametreleri, farklı yapıda yerel CAC'ların ağ içinde kullanımına izin verebilecek ve destekleyebilecek kadar esnek bir yapıda olmalıdır.



## **5 PNNI PROTOKOLÜNÜ İYİLEŞTİRME ÇALIŞMALARI**

Hızla gelişen ATM ağlarının ihtiyaçlarını karşılamak üzere, ATM Forum 1995 yılında PNNI v0.0 olarak da adlandırılan Interim Inter-Switch Signaling Protocol (ISSP) protokolünü piyasaya sürmüştür. Küçük çaplı ağ topolojilerinde kullanılmak üzere tasarlanmış olan bu sürüm, statik topoloji bilgilerini kullanmakta ve birçok ayarın kullanıcı tarafından yapılmasını gerektirmekteydi. Bu sebeplerden, bu geçici sürüm yerine, 1996 yılında ATM Forum amaçladığı gelişmiş yönlendirme ve işaretleşme özelliklerine sahip PNNI v1.0'ı tasarlamıştır. Gelişen piyasa şartları ve firmaların taleplerine cevap vermek için, PNNI v1.0 üzerine çeşitli eklentiler yapılmakta ve PNNI v2.0 üzerine çalışmalar devam etmektedir.

Bu bölümde, PNNI protokolünün başarımını etkileyen faktörler ve PNNI konusundaki ileriye yönelik çalışmalar ele alınacaktır.

### **5.1 PNNI Protokolünün Başarımını Etkileyen Faktörler**

ATM Forum tarafından standartlaştırılan Özel Ağdan-Ağa Arayüz (PNNI), ATM ağlarına esnek ve ölçeklenebilir bir yönlendirme mimarisi sağlamaktadır. Temel olarak OSPF yönlendirmesini kullanan PNNI, garantili bant genişliği ve sınırlanmış gecikme gibi gerçek zamanlı uygulamaların gereksinimlerini de karşılamak amacıyla geliştirilmiştir. Ama bütün bu getirilerin sonucu olarak PNNI gerçekleştirilmesi oldukça karmaşık bir yapıya sahip olmuştur. ATM Forum PNNI standardını belirlerken, bir çok tasarım alanında yol göstermekle beraber özellikle PNNI uygulandığı ağlar üzerindeki başarımını büyük ölçüde belirleyebilecek tasarım konularını PNNI'ı geliştirecek olan firmalara bırakmıştır. Bu bölümde PNNI başarımını etkileyen faktörlerden bahsedilecektir.

### 5.1.1 Hiyerarşinin Etkileri

Bu bölümde, birçok topoloji parametresi tarafından tanımlanan düzenli bir hiyerarşik ağ düşünülerek, ATM PNNI yönlendirme şeması tarafından ağ düğümlerine yüklenen güzergah hesaplama yükü ele alınmıştır.

Felstain ve Cohen[14] çalışmalarında bir  $m$ -seviyeli PG'nin hiyerarşinin  $m$ 'inci seviyesinde bulunan bir PG olduğunu rekürsif olarak  $m$  defa PG'lere bölünebildiğini belirtmiştir. Buna benzer olarak bir  $m$  seviyeli LGN'nin de hiyerarşinin  $m$ 'inci seviyesinde yer alan bir LGN olduğunu belirtmişlerdir. Her mesaj tarafından yaptırılan güzergah hesaplamalarının toplam sayısı ağ topolojisinin ve mesaj en üst yönlendirme seviyesinin bir fonksiyonu olarak ifade edilmiştir. Aynı yolla, mesaj güzergahından tek bir  $m$ -seviyeli PG geçirilerek (herhangi bir  $m$  değeri için) uygulanan güzergah hesaplamalarının sayısı hesaplanmıştır. Bu  $m$ -seviyeli güzergah hesaplamalarının uygulandığı düğümlerin sayısı da ( $m$ -seviyeli fiziksel sınır düğümleri olarak adlandırılır) elde edilmiştir.

Bu sonuçlar kullanılarak, birçok durumda  $m$ -seviyeli fiziksel sınır düğümlerindeki tek bir mesaj tarafından yaptırılan ortalama güzergah hesaplamaları sayısının,  $m$ 'e bağlı üssel olarak çoğalacağı gösterilmiştir. Ayrıca, mesaj güzergahında bulunan ama yönlendirme sürecine katılmayan düğümler ile  $m$ -seviyeli fiziksel düğümlerin birbirine oranının da  $m$  ile doğru orantılı olarak arttığı gözlemlenmiştir. Son olarak, birçok pratik çağrı dağıtımı durumunda,  $m$ -seviyeli fiziksel sınır düğümü tarafından uygulanan güzergah hesaplamaları sayısının da  $m$ 'e bağlı üssel olarak artabileceği belirtilmiştir. Bu, güzergah hesaplama yükünün ağ düğümleri arasında dengesiz olarak dağıtılabileceği anlamına gelir.

Üst seviyeli PG çiftleri arasındaki bağların uç noktalarını birleştiren, yüksek seviyeli fiziksel sınır düğümlerinin, güzergah hesaplarıyla fazla yüklenmesi ihtimali olduğu belirtilmiştir. Büyük hiyerarşik ağlarda bu aşırı yüklenmiş düğümler PNNI yönlendirme mekanizmasındaki görevlerini yerine getirmekte başarısız olabilirler.

Bu sorunun aşılmasına yönelik olarak Felstain ve Cohen çalışmalarında, ağ düğümleri içindeki hesap yükünü dengeleye yönelik bir şema önermişlerdir. Bu şemanın ardında yatan ana fikir, aşırı yüklü düğümlerdeki güzergah hesaplarını az

yüklü düğümlere aktararak, az yüklü düğümlerdeki boş geçen süreleri değerlendirmektedir.

### 5.1.2 Topoloji ve Bağ Durum Metriklerinin Toplanması

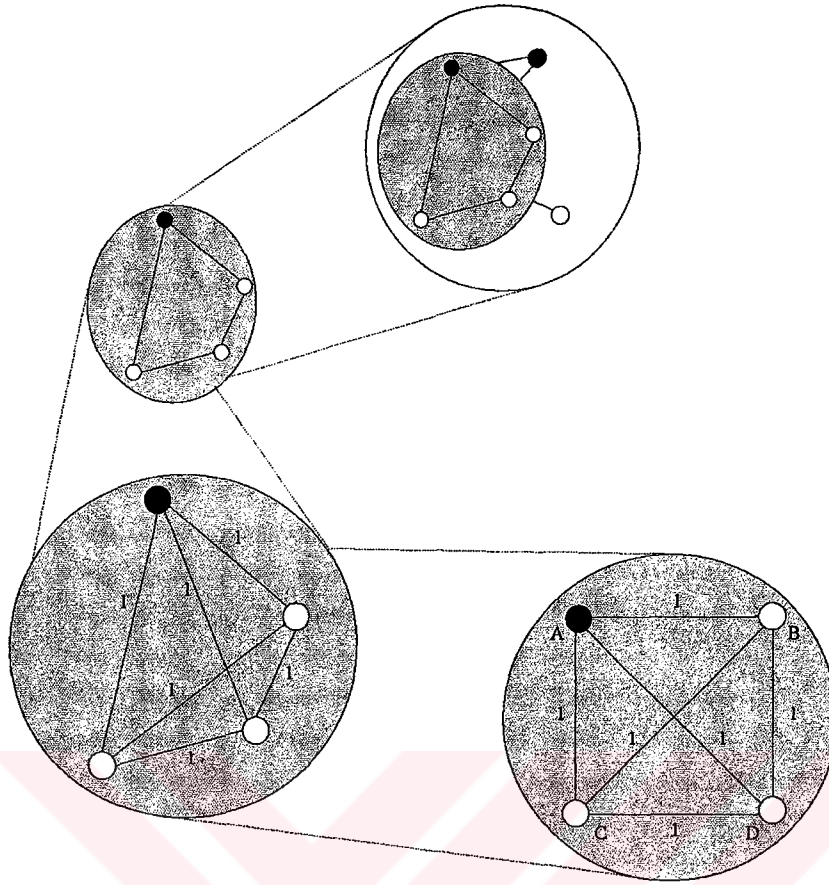
Modern yönlendirme teknolojisinin temeli TOS adı verilen bir dizi kritere dayanmaktadır. TOS, servis kalitesi faktörlerine (QOS) işaret eden bir Internet terimidir. Bu faktörler ağ yöneticileri ve kullanıcıları tarafından belirlenirler ve gecikme, girdi miktarı ve güvenlik ihtiyacı gibi kriterler içerebilir. Talep edilen servisi karşılamak için Internet'teki bir güzergah ağların ve yönlendiricilerin yeteneğine bağlı olarak seçilir.

Bu teknik ağ durum makinesi olarak da adlandırılabilir. Bunun nedeni, TOS değerlerinin Internet'teki her iletişim bağına uygulanmasıdır. Bir bağ metriği, kaynak adresten hedef adrese giden güzergah üzerindeki bağ durum parametrelerinin (TOS'lara bağlı) toplamı olarak tanımlanır. PNNI'nın amaçlarından biri de düğüme, iki düğüm arasındaki "optimum" (en kısa) güzergahı seçmesi için bu metrikleri tanıtmının yollarını bulmaktır. Düğümler arasındaki en kısa güzergahın uzunluğu, düğümler arasındaki uzunluk olarak bilinir.

Tipik büyük ağlarda, bir düğümden diğer düğüme birden fazla güzergah olacaktır. Dahası, ağ içindeki bu değişik kombinasyonlar tüm olası gösterimleri zor bir iş haline getirebilir. Ama, ağ içindeki düğümlerden biri (mesela bir PNNI mantıksal grup düğümü) bir şekilde bu metrikleri kısa bir özet şeklinde toplayıp sonra bu özeti tanıtır, görevin yerine getirilmesi kolaylaşır.[15]

Fakat, metriklerin toplanmasının getirdiği bu kolaylığının bir bedeli vardır. Birçok durumda elde edilen sonuç, en uygun güzergaha doğru gitmeyecektir. Bu sistemlerde, toplanmanın miktarı, toplanmanın doğruluğuna karşı tartılmalıdır.

Şekil 5.1'de, dört düğüm birbirine dört fiziksel bağ ile bağlanmıştır. Metrik toplanma ile ilgili noktalar oluşturulması için, şeklin alt kısmındaki topoloji, örgülü bir yapı ile değiştirilmiştir. A, B, C ve D düğümleri 6 bağ ile birbirlerine bağlanmıştır. Bu örnekte, ağ durum parametresi "1" her bağa atanmıştır.

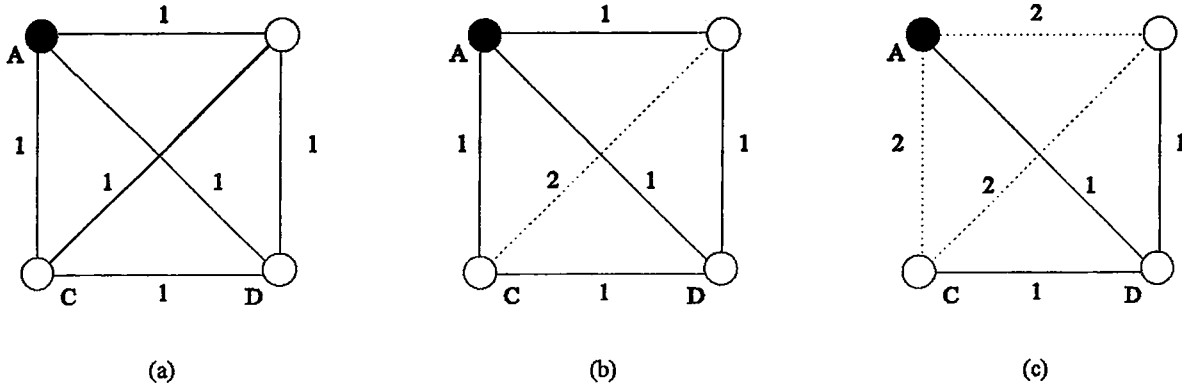


Şekil 5.1Bağ Metrikleri

#### 5.1.2.1 Metrik Toplanması

Şekil 5.2a'daki 6 adet bağa sahip olan tam örgülü ağ, bir alt graf olan şekil 5.2b'deki 5 bağlı ağda toplanmıştır. Şekil 5.2'deki noktalı çizgiler, bağların ağ mantıksal topolojisinden çıkarıldıklarını göstermektedir. Başka bir deyişle grafın dışına toplanmışlardır.

Metrik toplamada önemli amaçlardan biri, ufak gerilme faktörü olan tamamlanmamış bir alt graf oluşturmaktır. Gerilme faktörü, alt grafta bulunan uzaklığın orijinal graftaki uzaklığa oranıdır. 5 bağlı alt grafik için gerilme faktörü 2'dir ve mantıksal tamamen örgülü topolojinin korunması için 5 bağ kullanılmıştır. Burada orijinal graftan fark, B ile C arasındaki uzaklığın şimdi 2 olmasıdır. Fakat toplanma değişken sayısını azaltacaktır. Dahası, masrafı sebebiyle B ile C arasında fiziksel bağ konulmaması da tercih edilebilir.



(a) tamamen örgülü düğümler, (b) 5 bağla mantıksal olarak örülmüş ve gerilme faktörü=2 , (c) 2 bağla mantıksal olarak örülmüş ve gerilme faktörü=2

Şekil 5.2 Metrik Toplanması.

Şekil 5.2c'de bu ağ için metriklerin toplanmasının başka bir yolu gösterilmektedir. 3 bağlı toplanma halen tam bağlanabilirlik sağlamakta ve gerilme faktörü 2'de kalmaktadır. Şimdi A-C ve B-D arasındaki uzaklıklar 2'dir. Yine orjinal ağdan bir fark mevcuttur ve bu durum bu düğümler arasındaki trafiğe ve bunlar arasında direkt bağlar kurmanın getireceği masraflara bağlı olarak tercih edilmeyebilir.

PNNI, ağ yöneticisine fiziksel topolojide nasıl değişiklikler yapacağını söylemez fakat topolojiyi tanıtmak ve ağı etkin olarak kullanmak için bir araç verir.

ATM ağlarında ağ özelliklerinin düğümden düğüme değişmesi olasıdır. Bağlar aynı da olsa kullanımı farklı olabilir; bazı bağlar diğerlerinden daha fazla trafik taşıyabilir. Bu durum gecikme ve girdi miktarına dair değişik TOS değerlerinin oluşmasına sebep olabilir.

Şekil 5.3'de değişen bağ özelliklerine sahip metrik toplanma örnekleri gösterilmektedir. Şekil 5.3a'da tamamen örgülü bir ağ, Şekil 5.3b'de 5 bağlı toplanma gösterilmekte ve Şekil 5.3c'de ise 3 adet bağın toplanmasından oluşmuş minimum kapsayan ağaç temsil edilmektedir.

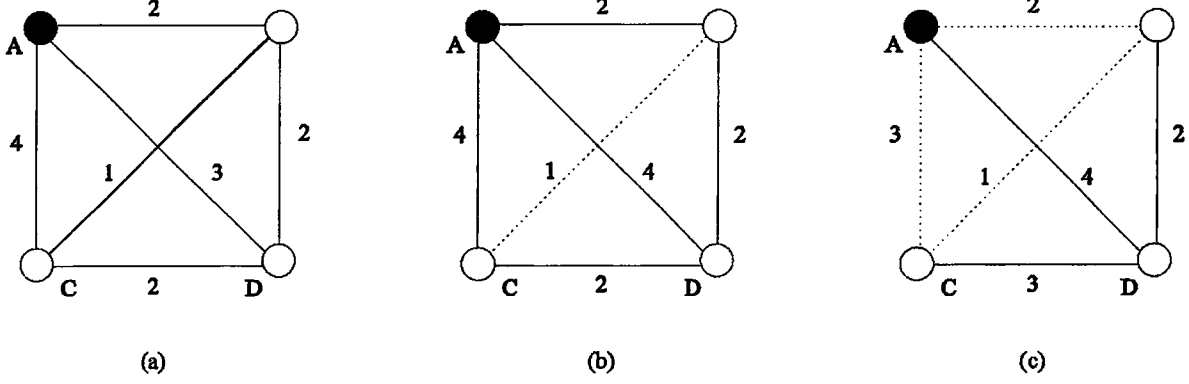
Bu örnekte, fiziksel olarak komşu düğümler arasında komşu olmayan düğümler üzerinden kısa güzergahlar bulunabileceği gösterilmektedir. A ve C fiziksel bağları arasındaki direkt güzergah 4'tür, fakat B üzerinden A ve C arasındaki birleşmiş güzergah 3'tür. Örneklerdeki gibi, bu durum eğer:

(1) düğüm A'nın C'ye olan bağı tıkalıysa

(2) bağ, düşük bit-oranında çalışıyorsa,

(3) metrik, bağdaki hata oranıyla ilgili olarak yüksek giderleri temsil ediyorsa

ve benzeri sebeplerden dolayı ortaya çıkabilir.



(a) tamamen örgülü ağ, (b) 5 bağlı, (c) 3 bağlı.

Şekil 5.3 Değişik bağ özelliklerinde metrik toplanması

Yönlendirme değişim bilgileri, bir PNNI sistemi içinde her düğümün PNNI topoloji durum elemanlarını (PTSE'ler) tüm ağ içinde, peer grup denilen belirlenmiş bir bölüme taşınını ile tanıtılır. Bilgiler içinde, düğüm kimliği ve bağlarının komşularına olan durumları vardır.

Bu bilgiler, peer grup içinde her düğümdeki topoloji veritabanının güncel kalması için kullanılır. Topoloji veritabanı PTSE'ler içindeki bilgilerin yansımasıdır ve peer grup içindeki herhangi bir düğümde başka bir düğüme güzergahı hesaplamak için gerekli olan bilgileri içerir. PTSE'ler bağ durum parametrelerini ve düğümsel durum parametrelerini içerirler.

Topoloji toplanma işlemi, hiyerarşik ATM ağlarında bilgi saklamak ve karmaşıklığı azaltmak için kullanılan bir mekanizmadır. Lee[16] incelemesinde, bu sorun ve bu sorun için tatmin edici bir çözüm bulmanın zorluğundan bahsetmiştir.

PNNI Yönlendirmede topoloji tanıtımları agresif olabilir; mesela tanıtılan bir grup metrik ve özellik, tüm parametreler mevcut olsa da yapılabilir değildir. Agresif tanıtımlarda çağrı kurulumu "bütçe sorununa" tabidir. Toplanmış metrik değerler, çağrı için kurulan bağlantı sırasında, çağrıyı başlatanın beklediğinden daha hızlı artar ve bağlantı kurulumu yetersiz ağ kaynaklarına bağlı olarak iptal edildiğinde kontrol

mesajları üretilir. Tutucu tanıtımlarda ise, çağrılarını desteklemek için, yeterli ağ kaynağı bulunduğu kaynak tarafından bilinmediği için, çağrılar uzak kaynak tarafından bloke edilirken ağ kaynakları yetersiz şekilde kullanılmış olabilir. Agresif ve tutucu tanıtımlar arasındaki farklılıkların incelenmeye devam edilmesi gerekmektedir.

Topolojideki simetri, kesinlikten ödün vermeden etkin bir topoloji toplamasına izin verir iki tip simetri vardır: Bağ simetrisi ve düğüm simetrisi. Bağ simetrisi ile, her bağa bağ durum parametrelerinin her iki yöndeki değerleri aynıdır. Düğüm simetrisinde de bağ simetrisi vardır ve buna ek olarak, bir tam örgülü gösterimde, tüm mantıksal bağlardaki bağ durum parametrelerinin değerleri de aynıdır. Verilen simetrik özellikli bir topolojinin birleştirilmesi için sadece  $O(M)$  karmaşıklığı gerekir (burada  $M$  topolojideki düğüm sayısıdır). Örneğin, Kapsayan Ağaç yaklaşımı,  $(M-1)$  mantıksal bağa sahip toplanmış ufak topoloji üretir. Kapsayan ağaç yaklaşımı, asimetrik özellikli topoloji oluşturmak için genişletilebilir. Bu konuda bazı ön çalışmalar [17]'de anlatılmıştır. Ne yazık ki, bu durum için özellik değerleri, tanıtılan mantıksal bağlar aslında bir kapsayan ağaç oluşturmazlar ve karmaşıklık  $O(M)$  ve  $O(M^2)$  arasında değişir.

Metrikler açısından topoloji toplanması simetrik olsa da olmasa da karmaşık ve kayıplara yol açan bir süreçtir. Metrik toplanması için Kapsayan Graf yaklaşımı, hassasiyet çok önemli olmadığı sürece daha uygun gözükmektedir. Hiyerarşinin alt seviyesinde cazip gözükken hiyerarşik topoloji toplanması, hiyerarşinin üst seviyelerinde hatalara sebep olabilir. Kapsayan graftan türeyen tutucu tanıtımlara izin verilirse, yönlendirme ve ağ kaynak yerleşimi büyük tehlike altına girebilir.

Düğüm simetrisinin, topoloji toplanmasında kolaylık sağladığı yaygın bir inanıştır, fakat günümüzün ATM ortamlarında yetersiz kalmaktadır. Bağ simetrisi, karmaşıklıkta marjinal bir artışla birlikte, daha kesin topoloji toplanması yolunda ileri bir adımdır. Fakat, asimetrik bağ parametreleri pratik kullanımda çok yaygın olmadığından asimetrik metrik ve özelliklerle topoloji birleştirmenin daha iyi yolları arzu edilir.  $O(M)$ 'den daha büyük olan herhangi bir karmaşıklık engellendiğinde, topoloji toplanmasının hassasiyeti bozulabilir. Hassasiyet arzu ediliyorsa, toplanma algoritmasının karmaşık olması ve toplanmış temsilin karmaşıklığının, sınır düğümleri sayısından daha fazla olması gerekir.

PNNI karmaşık düğüm temsili bir peer grup içindeki sınır düğümleri arasındaki bağlanabilirliği anlatmak için kullanılan esnek bir temsildir. İstisnaların kullanımı mevcut olanların yanı sıra gelecek topoloji toplanma yöntemleri için ileri adımlar sağlar.

### 5.1.3 Hiyerarşik Yönlendirme Algoritmaları

Yönlendirme algoritmaları iki ana sınıfa ayrılabilir: uzaklık vektörü algoritmaları ve bağ durum algoritmaları. Uzaklık vektörü algoritmalarında, yönlendiriciler hedef ağ başına bir giriş ile tablolar oluşturur. Bu tablolarda, güzergah maliyetleri ve hedefe bir sonraki adım saklanır. Bu tablolar, birbirine yakın komşular arasında değiş tokuş edilir; bu durum da güzergah maliyetleri hakkındaki bilginin yayılmasını yavaşlatabilir. Uzaklık vektörü algoritmalarında, bağ hatalarından oluşan bozulmalar kolaylıkla onarılamaz, bu durumda hesaplanan güzergahlarda döngüler oluşmasına neden olur. Bağ durum algoritmalarında, ağ yönlendiricileri kendilerine ait bağ maliyetlerini taşkın yöntemiyle ağa dağıtırlar. Güzergahtaki bir sonraki adım, her yönlendiricide tutulan tam topoloji veritabanını temel alınarak hesaplanır. Bağ durum algoritmaları, bağ hatalarının onarılmasında daha hızlı davranır.

Her iki algoritma sınıfı da, ölçeklenebilirlik problemi yaşar: uzaklık vektörü algoritmalarında düğümlerin sayısı ile doğrusal olarak tablo boyu büyür; bağ durum algoritmalarında bağlantı maliyeti, bağların sayısıyla büyür ve topoloji oluşturulması için boşluk gereksinimi de büyür. Ölçeklenebilirlik sorununun aşılabilmesi için hiyerarşik yönlendirme kavramı ortaya atılmıştır. Hiyerarşik yönlendirme ağ hakkındaki topoloji bilgilerinin düğümlerde çeşitli toplanma metodları kullanılarak, saklanmasını esas almaktadır. Bu da, saklanacak yönlendirme bilgi miktarını ve diğer düğümlerle değiş tokuş edilecek topoloji bilgisi miktarını önemli ölçüde azaltır.

Toplanma metodları, yönlendirme kararlarının alınmasında kullanılan bilgileri özetlediği için, bu kararların doğruluğunu etkiler. Bu durumun yönlendirme performansına olan etkisi birçok araştırmaya konu olmuştur.

Awerebuch, Du ve Shavitt[18] araştırmalarında, PNNI hiyerarşik yönlendirmesinin etkinliğinden bahsetmiştir. Farklı hiyerarşilere sahip ağların hiyerarşik yönlendirme performansları üzerinde çalışmalar yapılmıştır. Özellikle, global yönlendirme (tek

hiyerarşik seviyeli) ve hiyerarşik yönlendirme stratejisi, gerçekleştirilen bağlantılar için işaretleşme gecikmesi ve ağ girdi (debi) miktarı açısından kıyaslanmıştır.

Çalışmaların sonucu genel olarak göstermiştir ki, global yönlendirme stratejisine kıyasla hiyerarşik yönlendirmenin performansı daha iyidir; daha az depolama ve iletişim giderine fakat bununla birlikte daha fazla hesaplama giderine sahiptir. Bazı ağlar ve topolojilerde, hiyerarşik yönlendirme global yönlendirmeden bile daha iyi performans sağlar.

İncelenen başka bir konu da, aynı topolojiyi kullanan farklı hiyerarşilerin değişik performanslara sahip olmalarıdır. Yönlendirme performanslarını iyileştirmek için ağların hiyerarşisi üzerinde daha fazla çalışmaya gerek vardır. Bağ gider metriklerinin yönlendirme performansları üzerinde etkisi büyüktür. Çalışmalar sonucunda hiyerarşik yönlendirmede kullanılan üssel bağ gider metriğinin (bağ giderinin, bağın kalan bant genişliğinin üssel bir fonksiyonu olduğu durumda) büyük ağlar için etkin yönlendirme çözümü sağladığı ortaya çıkmıştır.

#### 5.1.4 QoS Garantisinin Sağlanması

Bir kullanıcı bilgisini transfer ederken ağdan, kaynaktan hedefe kadar olan güzergah boyunca QoS önlemlerine tabi olmasını isteyebilir (gecikme, maliyet, bant genişliği ve hücre kaybı gibi). QoS'u destekleyen bir ağ topolojisi QoS önlemlerinin bileşeni olan bağ metrik vektörlerinden oluşur. Bir güzergahın QoS önlemleri ya toplanabilir ya da minimum(maksimum) olabilir. Güzergahın QoS önlemleri, toplanabilir olduğu durumda, bu değer güzergah boyunca QoS metriklerinin toplamıdır. Minimum(maksimum) QoS önlemleri, talep edilen minimum(maksimum) QoS önlemlerini sağlayamayan bağları, güzergah hesaplanmaya başlamadan önce ihmal eder. [19, 20]

Toplanabilir QoS önlemleri birçok güçlük ortaya çıkarır: birden fazla toplanabilir kısıtın ( $\geq 2$ ) NP-karmaşıklığını yaratması güzergahın hesaplanmasını ve böylece geniş ağlar için izlenebilirliği güçleştirir.

Literatürde, bu çok kısıtlı problemin, bazı özel durum ve varsayımlarda polinomial zamanda çözülebileceği yer almıştır. Fakat, genelde NP-karmaşıklığındaki problemler polinomial zamana indirgenmesi pek mümkün değildir. Dolayısıyla,

çeşitli yaklaşık çözümler ve sezgisel yaklaşımlarla seçilecek güzergahın kısıtların tamamını sağlayıp sağlayamayacağını garanti etmeden polinomial zamanlı olarak hesaplanmasında kullanılabilir. Esasen sezgisel kurallar, birçok metriğin tek bir skaler büyüklük haline getirilmesini sağlar, (mesela efektif bant genişliği) ki bu da Dijkstra'nın ki gibi bir algoritmanın kullanılmasına izin verir.

Iwata[21], PNNI'la beraber çalışacak ve dolayısıyla PNNI işaretleşme ve topoloji değişim protokollerini kullanacak bir algoritma önermiştir. Önerilen bu yönlendirme algoritmasında, ilk etapta yeni bağlantının talep ettiği eşdeğer bant genişliğini destekleyemeyen tüm bağlar budanır. Daha sonra tek bir QoS parametresi seçilir ve bir veya daha fazla olası güzergah, seçilen metriği temel alan en küçük maliyetli kriter kullanılarak bulunur. Daha sonra, bu olası güzergahların QoS metriklerini uçtan uca sağlayıp sağlayamadığı kontrol edilir. Eğer kabul edilebilir bir güzergah bulunamazsa algoritma farklı bir QoS parametresi kullanarak yukarıdaki adımları bütün olasılıklar eleninceye kadar tekrar eder.

Eğer birçok kabul edilebilir güzergah bulunduyorsa, algoritma bir güzergah seçmek için yük dengeleme algoritmasını uygular ve bağlantı talebi ilerletilir. Ağ yöneticisi tarafından atanan yönetimsel ağırlık (AW), en kısa güzergahın hesaplanmasında kullanılabilir. Bu bağlamda hesaplanan güzergahlar ağ kaynaklarını optimal olarak kullanmayı amaçlar, fakat optimal çağrı bloke olma olasılığını oluşturmaz. Dolayısıyla, yönetimsel ağırlıktan başka kriterleri kullanan şemalar daha düşük çağrı bloke olma olasılığı sunmakla beraber, daha düşük kullanım etkinliğini de beraberlerinde getirirler.

#### **5.1.5 PNNI Protokolünün Ağa Getirdiği Ek Mesaj Yüğü**

Büyük peer gruplara sahip ATM anahtarlama sistemleri için kritik olabilecek topoloji bilgi paketlerinin işletim yükü, ağın dengesini bozar ve hata oluşumlarına sebebiyet verir. PNNI'nın oluşturduğu bu ek yükün, ağa yaptığı etkileri araştıran birçok inceleme vardır.[22]

Bu yükün önemli bir kısmını oluşturan topoloji bilgi paketlerinin (PTSP'ler) dağıtımını daha etkin bir şekilde dağıtımını gerçekleştirebilecek algoritmalar aranmaktadır. Topoloji bilgisinin dağıtılması için kapsayan ağaç algoritması, PNNI

standardı tarafından kullanılan taşkın mekanizmasına bir alternatif olarak önerilmiştir[23]. Bu çalışmada, Melez Kapsayan Ağaç (HST) algoritması sunulmuştur. Topoloji dağıtımı için kapsayan ağaç metodunun taşkın yöntemine göre daha üstün olduğu tartışılmaktadır. Topoloji dağıtımı için HST yaklaşımının diğer kapsayan ağaç metotlarına karşı en önemli avantajları: HST'nin ATM anahtar donanım modifikasyonlarını gerektirmemesidir, ve taşkın yöntemiyle topoloji dağıtımı algoritmalarını işletmeye devam eden standart PNNI düğümleriyle ortak çalışabilirler. HST, taşkın yöntemi algoritmasıyla işleyen PNNI düğümlerinin HST metoduna geçişin kolay olmasını sağlayan bir yapıya sahiptir. Sonuç olarak HST algoritmasında işleyen düğümler, taşkın yönteminde işleyen düğümlerle PNNI hiyerarşisinin herhangi bir seviyesinde ortak olarak çalışabilir.

Taşkın yöntemine karşı kapsayan ağaç yaklaşımının performans analizi topoloji dağıtımı için PNNI standardında gerçekleştirilmiştir. Bunu için topoloji dağıtımının hem taşkın yöntemi hem de kapsayan ağaç yöntemi üzerinden gerçekleştirilmesini sağlayan bir simulasyon yapılmıştır. Sonuç olarak PNNI topoloji dağıtımının dar bant genişlikli bağlar üzerinden yapılmasının çözümsüz olduğu sonucuna varılmıştır. Kapsayan ağaç yaklaşımı topoloji dağıtımı için daha az bant genişliğini kullandığından, kontrol trafiği için bağ kullanımı açısından taşkın yöntemine göre çok daha yüksek bir topoloji bilgisi güncelleme oranına sahiptir. Dolayısıyla, kapsayan ağaç yaklaşımı düğümlerde ağ topolojisi hakkında daha hassas ve doğru bir temsil sağlayacaktır.

#### **5.1.6 Crankback Mekanizması**

Crankback sağladığı faydalar yanında ağ üzerinde ek yükler de yaratır. Bu ek yük ise, çok fazla hiyerarşik seviyeye sahip geniş çaplı ağlarda, ağ hakkındaki QoS bilgileri doğru olarak özetlenemediği için, önemli boyutlara ulaşabilir. Felstaine[9] çalışmasında, hiyerarşik ATM ağlarında crankback oluşumunun önceden kestirimi için, belli bir QoS kotası aşılmadan crankback'i başlatan bir şema önermiştir. Bu şemanın temel amacı, PNNI'da bağlantı kurulumu gecikmesini azaltmak ve gereksiz işaretleme mesajları yüzünden oluşan iletişim yükünü ve işlemleri hafifletmektir.

Crankback tahmininde kullanılan bu şema, QoS kotasını ağ düğümleri arasında rekürsif olarak dağıtır. Bu çalışmada, QoS kotasının bölünmesinde kullanılan çeşitli

algoritmalar ele alınmıştır. Bu algoritmalarından bazıları deterministik, bazılarıda stokastik yaklaşımlara dayanmaktadır. Stokastik yaklaşımı kullanan algoritma, güzergahın kalan kısmı için gereken kotanın rassal yoğunluk fonksiyonunu kullanarak bağlantı kurulumdaki bir hata oluşma olasılığını belirler.

Bu algoritmaların başarımlar değeri, stokastik yaklaşımların daha iyi olduğunu ortaya koymaktadır. Fakat bu yaklaşımlar PNNI düğümlerinin durumlarına ait istatistiksel bilgileri tanıtmalarını zorunlu kılmaktadır, bu türde bir bilgi akışı ise PNNI standardının bir parçası değildir. Dolayısıyla uygulanabilmek için PNNI standardında bazı eklentilerin yapılmasına ihtiyaç duyarlar.

## 5.2 PNNI Protokolünde Son Çalışmalar ve Eklentiler

ATM Forum 1996 yılında PNNI'nin ilk sürümünü çıkardığından itibaren değışen şartlar ve gereksinimler yönünde çeşitli eklentiler ve düzeltmeler yapılmıştır. Bu eklentiler arasında en önemlileri ATM protokolünün mevcut ağ protokolleriyle özellikle popülaritesinden ötürü IP protokolüyle olan iletişimini ve bütünleşmesini kapsamaktadır. PNNI'nin ikinci sürümü, ilk sürüm itibariyle yapılan eklentiler ve firmaların önerdiği çeşitli ek hizmetlerin onaylanmasında sonra kullanıma sunulacaktır. Bu bölümde, PNNI v1.0 yapılan eklentiler ve firmaların ek hizmet önerileri ele alınmıştır.

### 5.2.1 Ağlararası Protokollerinin ATM üzerinden Yönlendirilmesi

Gelecekteki ağ topolojileri ATM anahtarları, yönlendiriciler ve hem diğeri ağ sistemlerine hem de ATM'e bağı olan konaklardan oluşacaktır[24]. Burada yönlendiricilerden kasıt, hem *klasik yönlendiriciler* hem de *sanal yönlendiriciler* olacaktır. Ağlararası güzergahların tutarlılığını sağlamak için, ATM ve diğeri ortamların kombinasyonunun, birbirinden farklı çeşitli ağ cihazlarıyla oluşturulması esastır. ATM ortamında çoklu protokolün yönlendirilmesi için üç ayrı yaklaşım vardır:

- Katmanlı yönlendirme,
- Geliştirilmiş Yönlendirmeli PNNI (PAR),

- Birleştirilmiş PNNI (I-PNNI).

Katmanlı yönlendirmede, ATM ağı içinde kullanılan yönlendirme protokolü (yani PNNI), yönlendiriciler arasındaki IP seviyesinde kullanılan yönlendirme protokolünden ayrıdır. Yönlendiricilerin ATM ağının iç yapısı hakkında bilgisi yoktur. Katmanlı yönlendirme IP seviyeli yönlendirmeyi ATM seviyeli yönlendirmeden ayırır. Katmanlama, kullanılacak SVC'lerin nasıl yönlendirileceğini (kurulacağını) bilmeyen geleneksel yönlendirme protokollerine destek verir. Sonuç olarak ortaya çıkan mantıksal ağ, IP yönlendirme protokolünde herhangi bir bağ katman ortamı olarak gözükecektir.

Bu yaklaşım yetersizdir, çünkü benzer fonksiyonlar zaten PNNI tarafından sağlanmaktadır. PNNI yönlendirmeyi OSPF, yönlendirme bilgi protokolü (RIP) ve sınır geçit protokolü (BGP) gibi mevcut yönlendirme teknolojileri ile birleştirip, bir peer yönlendirme mekanizması geliştirme sorununa yönelmeleri için ATM Forum grubundan iki alt grup oluşturulmuştur. Bu iki alt grup PAR ve I-PNNI diye adlandırılmıştır.

PAR, hala hem IP protokollerini, (OSPF ve RIP gibi) hem de ATM yönlendirme protokollerini (PNNI) kullanmaktadır. ATM ağına bağlı olan yönlendiriciler ATM yönlendirme protokollerine de katılırlar. Bunun amacı ATM'e bağlı yönlendiricilere ATM ağındaki diğer yönlendiriciler hakkında bilgi sağlamaktır (yönlendirme sunucuları da dahil). ATM ağının iç yapısı hakkındaki bilgi, ATM ağındaki SVC'leri düzenlemekte yardımcı olur.

I-PNNI, tüm yönlendiriciler ve ATM anahtarları arasında yönlendirme yapmak için tek bir yönlendirme protokolü kullanır. Tüm yönlendirme alanı içinde tek bir yönlendirme protokolü kullanılarak, uçtan-uca yönlendirmenin gerçek ağ topolojisine bağlı kalması ve PAR'daki gibi SVC'lerin oto-konfigürasyonu sağlanır. Sadece bir yönlendirme protokolünün yönetimine ve konfigürasyonuna gereksinim duyulur.

PAR ve I-PNNI'nin temel olarak ayrıldığı nokta PNNI'nin IP yönlendirme paketlerinde kullanılmasıdır[25]. PAR, ATM ağının sonundaki yönlendirmenin birleştirilmesi ile ilgilenirken, I-PNNI tüm birleşik topoloji ağını bir I-PNNI yönlendirme ağı olarak görür. PAR ile ilgili sorunlar arasında, ATM ağının diğer ağlarla nasıl temsil

edileceđi, kurulu SVC'lerin nasıl temsil edileceđi ve/veya toplanacađı ve IP paketlerinin nasıl yönlendirileceđi sayılabilir. PAR'da bu problemler ATM ađı sebebiyle sınırlanmıřtır. Halbuki, I-PNNI ile yönlendirme ve düđüm/bađ temsil ve toplanması klasik yönlendiricileri içerecek řekilde genişletilebilir.

#### 5.2.1.1 Geliřtirilmiř Yönlendirmeli PNNI

Geliřtirilmiř Yönlendirmeli PNNI (PAR), ATM ortamı üzerinden çoklu protokol yönlendirmesi yapmada kullanılan bir yöntemdir[26, 27]. Diđer internet seviyesi yönlendirme protokolleri (OSPF ve RIP) arasında çalıřtıđı gibi, bu yönlendirme protokolleriyle ATM arasında da çalıřabilir (ATM Sınır Yönlendiricilerinde PNNI kullanılır.). ATM sınır yönlendiricilerinin PNNI'a katılma sebebi, birbirlerinin yerini tespit edebilmek, bir ađ paylaşımı kurmak ve diđer ađlarla ATM arasındaki bakım iřlerini kolaylařtırmaktır.

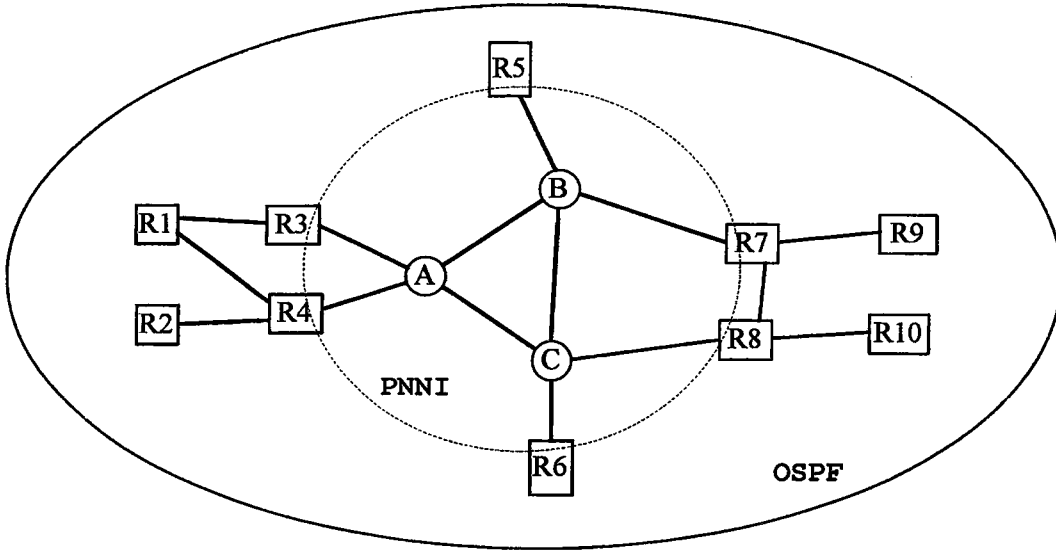
PAR kullanıldıđı zaman, tüm yönlendiriciler IP yönlendirme protokolünü çalıřtırır (örnek olarak: OSPF, RIP, EGP, BGP) ve bu yönlendirme protokolünün sonuçlarını normal IP paketlerini gönderme iřleminde kullanır. Tüm ATM anahtarları ise, standart PNNI yönlendirmesini kullanır. ATM arayüzlerine sahip olan bu yönlendiriciler, PNNI'ı çalıřtırır.

PAR'ı kullanan ATM'e bađlı yönlendiriciler, PNNI'a az sayıda ekstra bilgi verirler. Bu bilgiyle, ATM'e arayüzleri olan yönlendiriciler, ATM ađı içinde diđer düđümlerin nerede bulunduklarını öđrenir. PNNI'ın kullanılması bu yönlendiricilere, ATM ađının durumu hakkında detaylı ve dahili bir bilgi sađlar. Ayrıca özel olarak, ATM ađı içindeki yönlendiricilerin arasındaki SVC'lerin bakımını kolaylařtırır.

řekil 5.4'de kolay bir ađ örneđi gösterilmiřtir. řekilde, 10 adet yönlendirici (R1'den R10'a kadar adlandırılmıř) ve 3 ATM anahtarı A, B, C bulunmaktadır. Yönlendiriciler arasındaki yönlendirme, ATM ađı içerisindeki yönlendirmeden bađımsızdır. A, B, C ATM anahtarları arasında kullanılan yönlendirme protokolü, yönlendiriciler arasında kullanılan yönlendirme protokolünden farklıdır.

Anahtarlar ve yönlendiriciler arasında iliřimi sađlamak amacıyla kullanılan, yönlendirme sunucuları ve uç iletim aygıtları řekilde gösterilmemiřtir. PAR, ATM'e

bağlantısı olan yönlendiricilerin (yönlendirme sunucularının), ATM PNNI yönlendirme protokolüne katılımına ihtiyaç duyar.

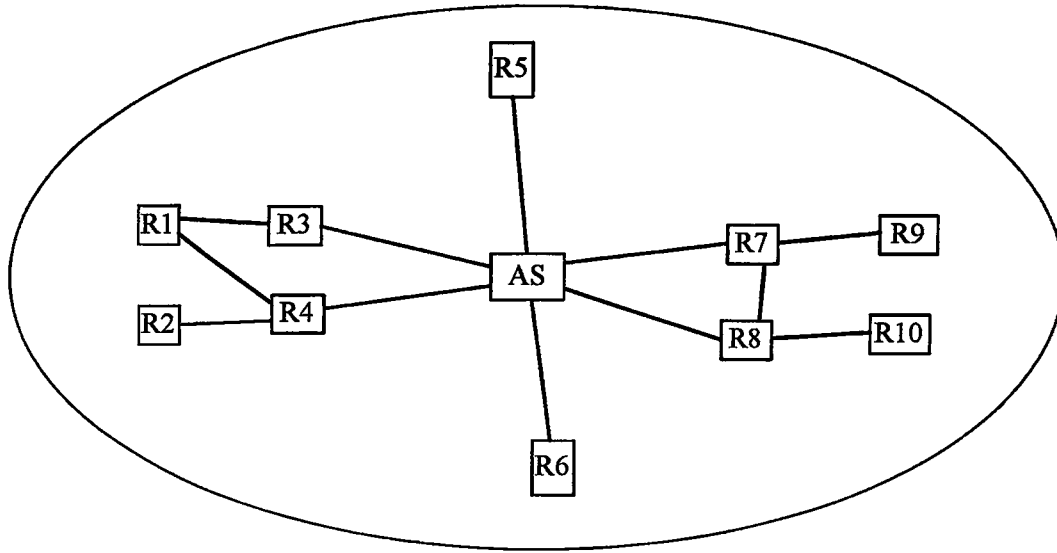


Şekil 5.4 PNNI ve OSPF'yi kullanan Ağ

Şekil 5.4'deki örnekte, 10 adet yönlendirici IP yönlendirme protokolünü kullanmaktadır (örnek olarak: OSPF seçilmiş olabilir). ATM anahtarları ve ATM ağına direkt olarak bağlantısı olan yönlendiriciler, PNNI'yi kullanmaktadır. Bu yöntemle, ATM'e direkt arayüzlerle bağlantılı olan yönlendiriciler, şekil 5.4'de gösterilen tam topolojiyi algılamış olur.

İki yönlendirme protokolünün (IP ve ATM) işlemleri birbirinden bağımsız olarak tutulur. Yönlendiriciler ATM ağı içindeki VC'leri, yönlendirme bilgilerinin değiş tokuşu ve paketleri göndermek için kullanabilirler. Yönlendiriciler arasında bir yönlendirme protokolü çalıştırmanın gerek olmasının sebebi, ATM ağının neleri yapabildiğini ve karakteristiklerini göstermektir. Eğer o an, ATM ağını geçen herhangi bir SVC kurulmamışsa, ATM ağı üzerinden paketlerin gönderilebilmesinin mümkün olduğu, yönlendiricilerden oluşmuş ağa bildirilmelidir. Burada kullanılabilecek iki yöntem vardır:

İlk yöntem, şekil 5.5'de gösterildiği gibi, mantıksal bir "ATM sahte düğümü" kullanmaktır. Burada, ATM ağı, yönlendiricilere ATM arayüzleri aracılığıyla bağlanmış bir mantıksal ATM düğümü olarak gösterilmektedir. IP yönlendirme protokolü ise, AS düğümünü normal bir yönlendiriciymiş gibi görür.

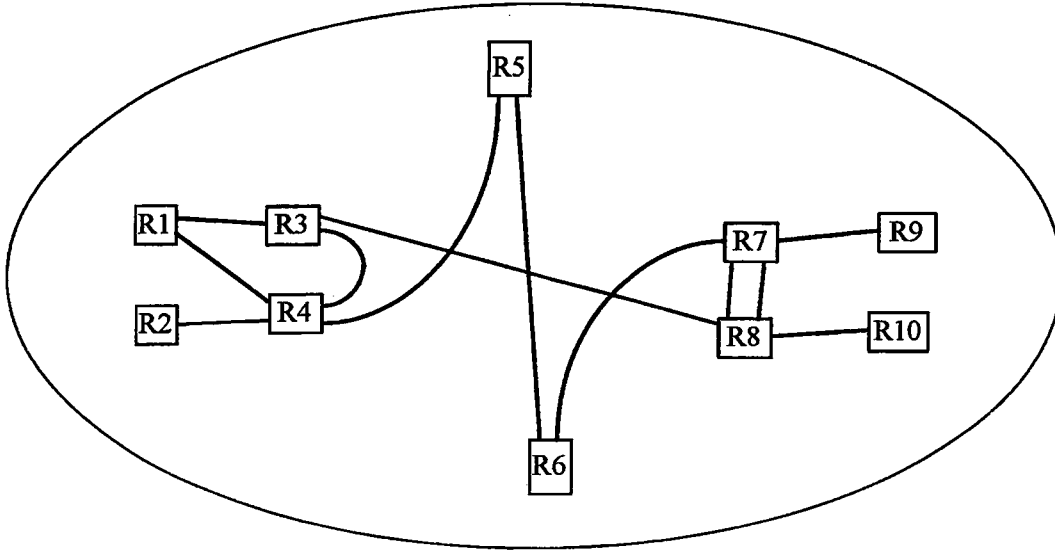


Şekil 5.5 Sahte Düğümlü Mantıksal Yönlendirici Topolojisi

ATM sınır yönlendiricilerinde bir tanesi, bu mantıksal düğümü temsil etmek için seçilir. Bu seçim manuel konfigürasyon ya da seçim algoritması kullanılarak yapılabilir. Bundan sonra, belirlenmiş yönlendirici diğer ATM'e bağlı yönlendiriciler arasında, OSPF LSA'ları iletmekten sorumlu olur. Sahte düğüm sadece, ATM'e bağlı olmayan yönlendiriciler arasında yönlendirme güzergahlarının ara düğümü olur.

ATM'ye bağlı olmayan yönlendiriciler, ATM veya PAR'dan habersiz topoloji içinde yönlendirmeye devam edebilirler. Fakat ATM'ye bağlı yönlendiricilerin, sahte düğümün ATM ağının bir temsili olduğunu bilmeleri ve böylece belirlenmiş düğümden sonraki adımı hesaplayıp, bu yönlendiriciye SVC yoluyla paketler göndermeleri gerekir.

İkinci yöntem ise, yönlendiricilerin, ağın başlatılması aşamasında henüz herhangi bir trafik oluşmadığında, hemen kendi aralarında, bir grup "varsayılan SVC" belirlemeleridir. SVC'lerin kurulması PNNI ve PAR'ın çalışmasıyla kazanılmış bilgilere bağlıdır ve ATM ağını kapsayan bir SVC grubu yaratılmasını sağlar. PNNI'a katılan herhangi X ve Y yönlendiricileri (aynı OSPF alanı içinde olan) için X'ten Y'ye direkt bir SVC olmasına gerek yoktur. Buna rağmen, X'ten Y'ye, ara adımlar olarak aynı OSPF alanı içinde başka yönlendiricileri kullanan bazı güzergahlar olacaktır.



Şekil 5.6 Varsayılan SVC'li Mantıksal Yönlendirici Topolojisi

Varsayılan SVC yaklaşımı, (Şekil 5.6) ATM ağını sınır yönlendiricileri arasında bir grup SVC olarak gösterir. Bağ tanıma açısından sahte düğüm ile aynı olduğundan aynı sorunlarla karşılaşılabilir.

PAR, PNNI içinde, ATM sınır yönlendiricilerin birbirinin yerlerini belirlemelerini, katman ağ kurulmasını ve en önemlisi ATM üzerinde ağlararası bakımını kolaylaştıran ve adres çözümlemesi gereğini ortadan kaldıran mekanizmalar sağlar. Bu modelde ATM sınır yönlendiricileri hem IP protokolünü kullanan klasik yönlendiricilerle yönlendirme bilgisi değişimine, hem de PNNI'yı kullanan ATM anahtarlarına katılır. PAR, ağlararası alt yapısı geliştikçe adım adım ATM'e geçişi sağlar; çünkü bu, klasik yönlendiricilerle bağlanmış ağlarda herhangi bir değişiklik yapılmasına ihtiyaç gerektirmez.

#### 5.2.1.2 Birleştirilmiş PNNI

Birleştirilmiş PNNI (I-PNNI), ATM üzerinden IP ortamında kullanılan ve PNNI yönlendirme protokolünün uzantısına dayanan bir yönlendirme protokolüdür[28, 29]. I-PNNI, PNNI protokolünün yönlendiriciler (güzergah sunucuları da dahil) ve ATM anahtarları tarafından kullanılmasını sağlar. PNNI, PNNI Helloları ve PNNI topoloji durum elemanları/PNNI topoloji durum paketleri değiştiren (PTSE/PTSP) anahtarlar ve yönlendiriciler arasında, ATM anahtarları arasında, ve ATM sınır yönlendiricileri arasında kullanılır.

Tüm yönlendiriciler ve anahtarlar PNNI topolojisinde düğüm olarak görünürler. I-PNNI, PNNI'yı kullanan fakat IP gibi ağlararası katman protokolü hakkında bilgisi olmayan ATM anahtarları ile uyumlu olacak şekilde dizayn edilmiştir. PNNI'nin esnekliği sayesinde bu uzantılar ve birleştirmeler kolaylaşır.

Hem ATM sistemlerinin, hem de IP sistemlerinin (yani IP alt ağlar) ulaşılabilirliği PNNI'ya tanıtılır. Fakat, farklı tipteki ulaşılabilirlikler birbirinden bağımsız olarak tanıtılırlar. ATM sistemlerinin ulaşılabilirliği PNNI'da belirlenen format ile tanıtılırken, IP alt ağlarının ulaşılabilirliği farklı tip uzunluk değeri-(TLV) kodlu I-PNNI eklentisi kullanılarak tanıtılır.

PNNI'yı IP yönlendirmesine genişleterek I-PNNI, PNNI'nin yeteneklerini tanıtır (QoS güzergahlarının uçtan uca tam ağ topolojisine bağlı olarak hesaplanması gibi). IP için, QoS desteği sağlamada bu özellik önemlidir. Buna ek olarak, SVC yönetimi için PAR özelliklerini tanıtır ve bu sayede yönlendiriciler ATM ağında birbirlerini bulup, SVC'lerin ağ içerisinde koordinasyonunu sağlar.

I-PNNI yönlendirici/anahtar topolojisindeki tüm düğümler arasında kullanılır. Bu sayede tüm düğümler topolojinin tamamını görebilirler. Fakat, anahtarların SVC'leri yönlendiriciler aracılığıyla yönlendirmeleri kabul edilemez. SVC bir yönlendiricide sonlanabilir ama başka bir düğüme ulaşmak için bir yönlendiricinin içinden geçemez. Bu engellemenin sebebi, yönlendiricilerin kendilerini *geçiş kısıtlamalı* düğümler olarak tanıtmalarıdır. Bir geçiş kısıtlamalı düğüm, düğümün geçiş için kullanılabileceği durumları belirten PTSE'ler tarafından geçişe açılabilir. Bu özellik, SVC'ler için olmasa da, bir düğümün IP paketleri gönderebileceğini belirten IP bilgileri içeren PTSE'ler tarafından kullanılır.

Helloların değiş tokuşu PNNI'da yapılandan farklı değildir. Her düğüm (yönlendiriciler de dahil) hello mesajlarını tüm komşularıyla değiştirir ve bu sayede düğüm komşuların kimliğini ve bağın durumunu belirler. Hello mesajları ayrıca komşunun aynı peer grupta olup olmadığını belirlemek için de kullanılır, böylece gönderilecek PTSE'lerin kapsamı kontrol altında tutulur. Tüm düğümler (yönlendiriciler de dahil) PTSE'ler kullanarak bağ durumlarını ilan ederler. PTSE'ler peer gruptan normal şekilde taşkın edilirler.

I-PNNI'ya katılan tüm düğümler ulaşabildikleri sistemlerin adreslerini açıklarlar. ATM anahtarları ulaşabildikleri ATM adreslerine (ATM özel adreslerinin ön eklerine bağlı olarak) ulaşılabilirliği açıklarlar ve benzer şekilde yönlendiriciler ulaşabildikleri IP adreslerine (IP adreslerinin ön eklerine bağlı olarak) ulaşılabilirliği açıklarlar. PNNI kullanan ATM'e bağlı yönlendiriciler, giriş ATM anahtarı tarafından açıklanmayacağı için kendi özel ATM adreslerine ulaşabildiklerini açıklarlar.

IP ve ATM adresleri ayrı ayrı taşınırlar. Bu ayrım sayesinde bağımsız adres atanması sağlanır ve ATM anahtarları IP ulaşılabilirlik bilgilerini yok sayarlar. IP adreslerine ulaşılabilirlik, normal PNNI kullanan anahtarlarca yok sayılacak TLV-kodlu uzantılarla açıklanır.

ATM ağı yoluyla yönlendiriciler arasında IP paketlerinin gönderilmesi için ağından geçmesi amacıyla SVC'lerin kullanılması gereklidir. SVC yönetimi sorunu, PAR'da karşılaşılan sorunla aynıdır. Bu sorun SVC'lerin nerede ve ne zaman kurulacağıdır. Olası yaklaşımların iki uç örneği aşağıda açıklanmıştır.

**Talep Edildiğinde Kurulum.** I-PNNI ile tüm yönlendiricilerin, ağ topolojisi hakkında tam bilgisi olacaktır. Bu sayede hiçbir SVC kurulu olmasa da uçtan uca güzergah ağ boyunca hesaplanır. ATM'ye bağlı bir yönlendiricinin ATM ağı üzerinden paketler göndermesi gerektiğinde, güzergah hesaplamaları, ATM ağı üzerinde bir sonraki yönlendiriciyi gösterir. Eğer bir SVC kurulu ise, yönlendiricinin tek yapacağı paketi göndermektir; aksi taktirde, talep edildiğinde SVC kurulur. Zamanla, paketlerin çoğu tarafından kullanılan genel SVC'ler yerleşecektir. Az kullanılan SVC'ler ya bir zamanlayıcı tarafından ya da zamana bağlı (yaşlandırarak) olarak kapatılırlar.

Bu yaklaşımın dezavantajı, belli bir sıçrama yönlendiricisine giden ilk paket, SVC'nin kurulmasını beklerken önemli bir gecikme yaşayabilir. Bu ayrıca, paketi göndermeden önce yönlendiricinin tampon kuyruğuna da yansıtılır.

**Önceden Kurulum.** Bu yaklaşım yönlendiricinin ağ başlatılmadan önce bir grup SVC kurmasını önerir. I-PNNI'dan elde edilen bilgiler ATM ağını kapsayan bir grup SVC'nin üretilmesini sağlar. Bu yaklaşımın dezavantajı, ATM ağındaki optimal

olmayan güzergahların kullanılmasıdır. İstenilen yönlendiriciye ulaşmadan önce, paketlerin birçok sıçramadan geçmeleri gerekir.

Bu iki yaklaşım olası çözümlerin iki uç noktasıdır. Fakat, önceki SVC'lerin, en iyi IP trafiği için, özellikle de kısa süreli IP akışları için kurulması arzu edilir. Bu sayede her hedef için bir güzergah mevcut olacaktır. Uzun süreli akışlar ve QoS trafiği için en iyi güzergahın talep üzerine kurulması muhtemeldir.

Bir peer grup içindeki tüm yönlendiriciler ve anahtarlar peer grubun tam topolojisini elde ederler, bunun içinde her düğüm yoluyla hangi adres örneklerinin ulaşılabilir olduğu bilgisi de vardır. Buna ek olarak anahtarlar, yönlendiricilerin gönderdiği SVC'ler için geçişe izin olmadığını belirten bilgilere de sahiptirler. Bu bilgi, yönlendirici ve anahtarların çağrı ve paketleri doğru olarak yönlendirmesine ilişkin detayları kapsar.

Kullanılacak güzergah hesaplamaları servise bağlı olarak değişebilir. Bu servisler arasında;

Adım adım yönlendirme kullanan en iyi çaba IP paketleri

Adım adım yönlendirme kullanan QoS-özel IP paketleri ve

Güzergah kurulumu kullanan QoS-özel IP akışları

I-PNNI, IP trafiğinin ideal şekilde sağlanması için adım adım yönlendirme kullanır. Diğer tüm bağ durum yönlendirme protokollerinde olduğu gibi bir peer grup içindeki yönlendiriciler, yönlendirme döngülerini önlemek için tutarlı yönlendirme hesaplamaları yapmak zorundadırlar. I-PNNI, en iyi çaba IP paketlerinin standart güzergah hesaplama yönetimi için bilinen Dijkstra hesaplamalarını önermektedir. I-PNNI'nin potansiyel faydalarından biri de IP için QoS-duyarlı bir yönlendirme protokolü sağlamasıdır.

I-PNNI, birleşik QoS yönlendirme protokolünü kullanarak, klasik yönlendiricileri ve ATM anahtarlarını birleştirmeyi vaat eder. Başka bir çözüm protokolüne artık gerek kalmayacağından bu yaklaşım, diğerlerine kıyasla performansı önemli oranda artırır. PNNI'nin yapısında var olan esneklik üzerine inşa edildiğinden büyük ağlara uyum

sağlayabilir. Fakat, bu alanda kullanılan yönlendirici yazılım geliştirme işlemlerine gerek duyar.

Sonuç olarak, ATM ağları büyüdükçe ve IP katmanının konfigürasyonu zorlaştıkça, PAR devreye sokulabilir. PAR, SVC yönetimini kolaylaştırırken ağ ihtiyaçlarını sağlamak için (SVC talepleri) yeni güzergahlar (SVC'ler) tanıtmaya yolunda esneklik sağlar. ATM ağının büyüklüğü arttıkça, uçtan uca QoS yönlendirmenin gerektiği ortamlar için veya yeni ATM-merkezli ağlar için I-PNNI en mantıklı seçim olabilir.

### **5.2.2 Genel Uygulama Aktarımı**

Genel Uygulama Aktarımı (GAT) [30] PNNI v1.0 için opsiyonel bir eklentidir; şu an iki işlevi desteklemektedir:

Ortak-çalışmayı desteklemek için, bir PNNI ağı üzerinden PNNI olmayan işaretleşme protokolü bilgilerinin saydam olarak aktarılmasını sağlar. GAT tarafından desteklenen PNNI olmayan işaretleşme protokollerine örnek: ITU-T, ISUP ve FRF.10.1'dir.

GAT'ın sağladığı fonksiyonlar: Organizasyona Özel Bilgi Aktarımı ve Protokol Beraber Çalışma Bilgisi Aktarımıdır.

#### **Organizasyona Özel Bilgi Aktarımı**

GAT bilgi elemanı sayesinde, organizasyona özel bilgiler bir PNNI ağı üzerinden nakledilir. Bunlar ağa özel özellikler ya da satıcının tescil bilgileri hakkında özellikler olabilir. Bu, GAT bilgi elemanının içinde bulunan Organizasyona Özel Uygulama Tipi değeri kullanılarak elde edilir.

#### **Protokol Beraber Çalışma Bilgisi Aktarımı**

ATM ağlarının kullanımı arttıkça ses veya diğer ATM olmayan servisleri sağlamak için bu ağları kullanma eğilimi artmaktadır. Bu yüzden, PNNI işaretleşmesini diğer işaretleşme protokollerini destekleyecek şekilde genişletmek arzu edilebilir.

GAT bilgi elemanı sadece yukarıda bahsedilen fonksiyonları gerçekleştirmekle kalmaz. Bu özelliklerin desteklenmesi için gereken bilgilerin, birbirleriyle çalışan fonksiyonlar arasındaki düğümler tarafından değil de, sadece birbirleriyle çalışan

fonksiyonlar tarafından işlenmesi ve anlaşılması gerekir. Bu yüzden bu PNNI olmayan özelliklerin her birinin desteklenmesi için yeni PNNI bilgi elemanlarının tanımlanması yaklaşımı arzu edilmez. Bunun yerine GAT bilgi elemanı şekil 5.7’de gösterildiği gibi, ortak çalışan işlevler (IWF’ler) kullanarak, diğer protokollerden gelen bilgileri (bilgi elemanları veya parametreleri) PNNI ağı üzerinden bir diğer protokole, yine IWF’leri kullanarak saydam olarak geçirmek için kullanılır.



Şekil 5.7 Beraber Çalışma Mimarisi

### 5.2.3 PNNI İşaretleşmesine Güvenlik Eklentileri

Güvenlik servisleri ATM Güvenlik Ajansı tarafından sağlanır. ATM Güvenlik Ajansı, ATM Forum Güvenlik Tanımında ana hatlarıyla anlatılan ATM güvenlik servislerinin bakımı ve kullanılması için gereken yöntemleri sağlayan soyut bir nesnedir[31].

ATM Forum güvenlik ajansı mevcut PNNI v1.0 işaretleşme prosedürleri üzerinde ya direkt ya da dolaylı etkileri olan bir dizi güvenlik servisi ve mekanizmalarını verir. Güvenlik Ajansının fonksiyonları 4 parçaya ayrılmıştır. Bu parçalardan SAsmc, Güvenlik Mesajı Değişimini yerine getiren Güvenlik Ajansının fonksiyonlarını içerir. SAsmc fonksiyonları, mevcut PNNI v1.0 işaretleşme prosedürlerinden önce, esnasında ve sonrasında uygulanması gerekenler olmak üzere üç ana tip altında sınıflandırılabilirler[32].

### 5.2.4 Güzergah ve Bağlantı İzleme Eklentisi

Güzergah ve Bağlantı İzleme özelliği, eski ve yeni bağlantıların geçtiği mantıksal düğümler ve mantıksal bağları çabuk ve etkin olarak belirlemek için kullanılan kontrol mekanizmaları sağlar[33]. Bu bilgi tipik olarak ağ yönetiminde, özellikle de ağ hatalarını ayıklamada kullanılır. Güzergah İzleme ve Bağlantı İzleme iki ayrı özelliktir. Birbirleriyle herhangi bir etkileşimleri yoktur . Hem Güzergah İzleme hem de Bağlantı İzleme için; izleme, izleme kaynak düğümünden başlar, halbuki bağlantı,

izleme kaynak düğümünden yukarı doğru başlatılmıştır. Ayrıca izleme, izleme hedef düğümünde sonlanır, halbuki bağlantı, izleme hedef düğümünün ötesine kadar ilerleyebilir.

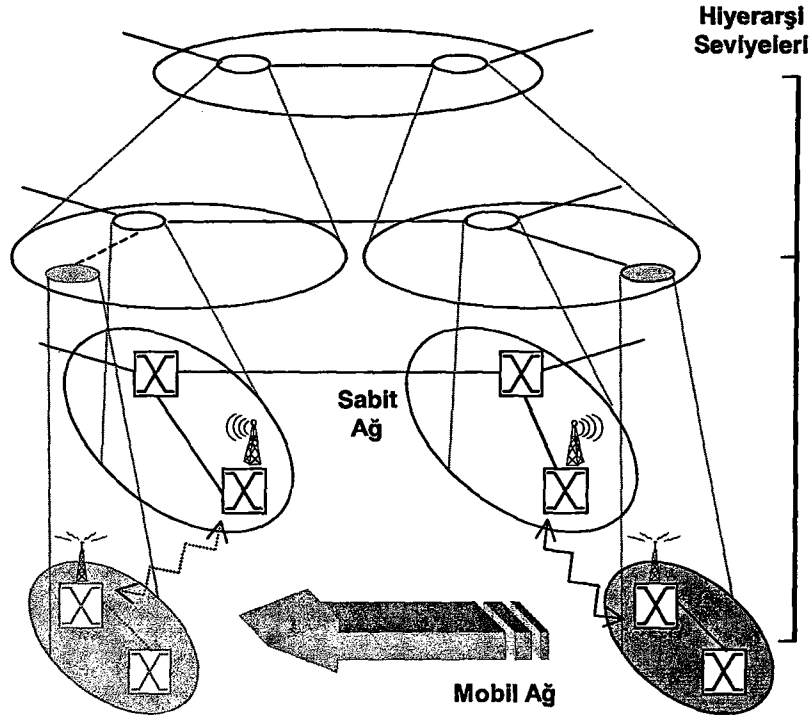
Güzergah İzleme özelliği kurulum aşamasında yeni bağlantılar için kullanılır. İzleme, ağ yönetimi tarafından yeni bağlantıların kullanacağı güzergahları belirlemek amacıyla başlatılabilir. Alternatif olarak, bunlar başka amaçlar için, mesela, bağlantı kurulum sorunlarını gidermek için izleniyor olabilir. İlk bahsedilen durumda, bağlantı asla kurulmaz; izleme hedef düğümüne ulaşıldığında, bağlantı gerçekleşeceği yerde yokolabilir. Güzergah İzleme özelliği yeni mesajlara ihtiyaç duymaz ama yeni İzleme geçiş listesi bilgi elemanının, SETUP, ADD PARTY, CONNECT, ADD PARTY ACKNOWLEDGE, RELEASE COMPLETE, DROP PARTY ve ADD PARTY REJECT dahil birçok PNNI işaretleşme mesajına eklenmesini gerektirir. Güzergah İzleme özelliği standart PNNI ve bağlantı kontrol prosedürlerinden başka yeni prosedürler de gerektirir.

Bağlantı İzleme özelliği, zaten kurulu olan bağlantılardan bilgi toplamak için kullanılır. Bağlantı İzleme özelliği iki yeni mesaj; TRACE CONNECTION ve TRACE CONNECTION ACKNOWLEDGE ve bir yeni bilgi elemanı gerektirir; İzleme geçiş listesi bilgi elemanı. Bağlantı İzleme özelliği mevcut çağrı ve bağlantı kontrol mesajları ve prosedürleri ile etkileşimde bulunmaz.

### 5.2.5 PNNI Mobillik Eklentisi

Bu eklentide ele alınan ağ mimarisi, mobil bir ATM ağıdır (*mobil ağ* olarak adlandırılır); bir ATM ağlararası altyapısına (*sabit ağ*) bir veya daha fazla kablosuz bağlantıyla bağlanan, bir veya daha fazla ATM anahtarından oluşur[34].

Mobillik eklentisi sayesinde, bir mobil ağ, sabit bir ağın PNNI yönlendirme hiyerarşisine, iki ağın bağlı olduğu noktalardan katılabilir. Sabit PNNI düğümlerinin aksine, mobil bir ağın bağlantı noktaları zamanla değişir. Bu eklentiler sayesinde, sabit ağa ait bir peer gruba üyeliğini dinamik olarak değiştirerek, iki peer grup arasındaki sınırı geçebilir (Şekil 5.8). Bu şekilde, iki mobil ağın uç sistemleri arasındaki geçişlilik ile, mobil ağdaki uç sistemlerden sabit ağa (ya da tam tersi) yönlendirme mümkündür.



Şekil 5.8 Erişim Noktası Anahtarı Değişen Mobil Ağ

Bağlantı hızı bu eklentilerde ele alınmaz. Eğer mobil ağ, bir erişim noktası anahtarı ile bağlantıyı kaybederse, ağlar arasındaki tüm bağlantılar uç sistemler tarafından veya ağların sağladığı diğer otomatik mekanizmalar tarafından tekrar kurulmalıdır.

#### 5.2.5.1 Mobil Ağ, Mobil Anahtar, Erişim Noktası Anahtarı ve Mobil Bağ

Burada kullanılan *mobil ağ* terimi ile ,birbirleriyle ilişkili olarak hareket etmeyen bir grup mobil anahtar kastedilmektedir. Mobil ağın anahtarları, kablolu ya da kablosuz araçlar ile birbirleriyle bağlanabilir.

*Mobil anahtar*, mobilite eklentilerindeki kapasitelere sahip ve mobil olacak şekilde konfigüre edilmiş anahtar sistemidir. Bir mobil ağda, bir ya da daha fazla mobil anahtar olmalıdır.

*Erişim noktası anahtarı*, sabit bir ağa ait olan ve bir mobil anahtar ile kablosuz bağ kurma kapasitesine sahip anahtar sistemidir.

*Mobil bağ*, bir mobil anahtar ile erişim noktası anahtarı arasındaki bağlantıdır. Mobil bağın bağlantı kapıları mobilite özelliğini destekleyen kapılar olarak konfigüre

edilmelidir. Bir mobil bağ, ya fiziksel bir bağ ya da sanal güzergah bağlantısı olabilir.

*Sabit ağ*, bir veya daha fazla mobil olmayan anahtardan oluşur.

#### **5.2.5.2 Mobil Mantıksal Grup Düğümü**

Bir peer grubun erişim noktası anahtarları içinde hareket eden bir mobil ağ, PNNI v1.0 yönlendirme ile ele alınabilir. Fakat bu çözüm büyük sayıdaki mobil ağlar ve büyük sayıdaki erişim noktası anahtarlarına uyarlanamaz.

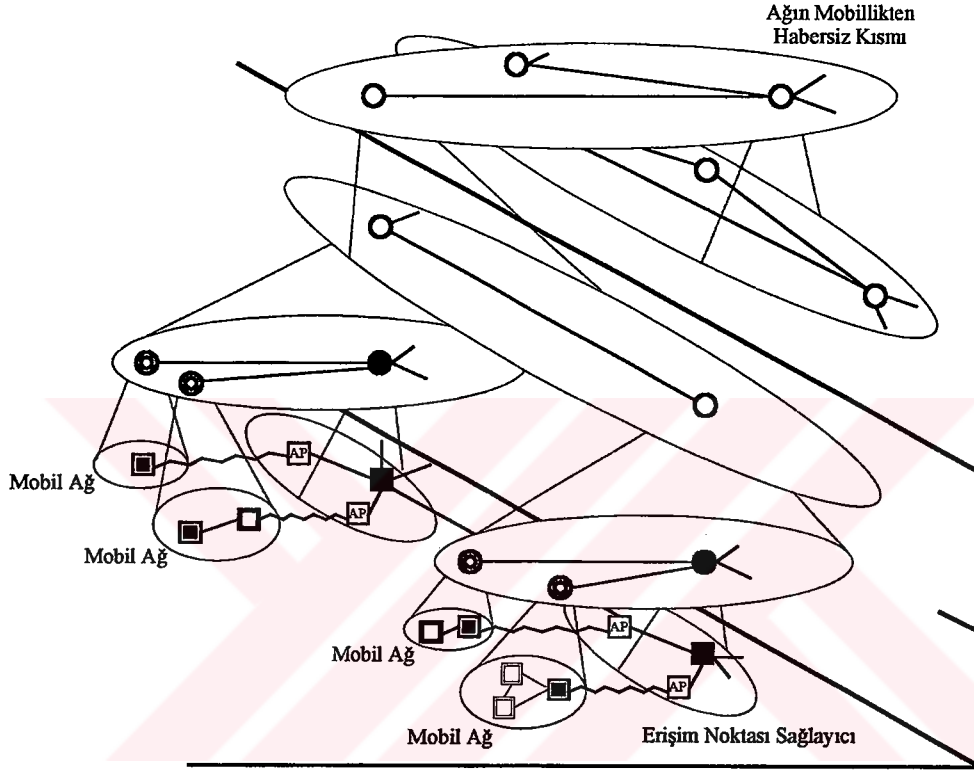
Mobillik eklentileri sayesinde, her mobil ağ kendi PNNI hiyerarşisini kurabilir ve sabit ağ hiyerarşisi içine mobil mantıksal grup düğümü şeklinde yerleştirilebilir. Bir mobil mantıksal grup düğümü, üyeliğini bir peer gruptan diğerine dinamik olarak değiştirme kapasitesine sahiptir. Bir mobil mantıksal grup düğümünün, sadece mevcut erişim noktası anahtarlarından birinin baba peer gruba katılmasına izin verilir.

Mobil ağa ait bir sınır düğümünün, bir veya daha fazla erişim noktası anahtarına, bir veya daha fazla aktif mobil dış bağı olabilir. Sınır düğümü, bir dış düğümsel hiyerarşi kurmak için erişim noktası anahtarından alınan düğümsel hiyerarşi listesini kullanır. Bu dış düğümsel hiyerarşi listesi, erişim noktası anahtarındaki mevcut konak peer gruplarını içerir. Daha sonra bir dış hiyerarşi listesi peer grup içinde kaynak sınır düğümü tarafından taşkın edilir ve sonunda peer grup liderine ulaşır. Her peer grup içinde ve mobil ağın hiyerarşisinin tüm seviyelerinde, kendi peer grubundaki düğümler tarafından tanıtılan birçok dış düğümsel hiyerarşi listesinin arasından birini seçmekten peer grup lideri sorumludur. Seçilen dış düğümsel hiyerarşi listesi sonra ilgili mantıksal grup düğümü tarafından, hiyerarşinin diğer seviyesine taşkın edilir. Hangi konak peer grubun katılacağı kararı, verilen mobil ağdaki en üst seviye peer grubun lideri tarafından verilir, bu düğüm mobil mantıksal grup düğümünü başlatır.

#### **5.2.5.3 Ağ Yapısı**

Ağ, mobil ağlar, erişim noktası sağlayıcıları ve sabit ağın kalan kısmı olarak üç parçaya bölünmüştür: (Şekil 5.9). Mobil ağ, erişim noktası sağlayıcısının ağ

hierarchyesine, tek bir mantıksal grup düğümü olarak birleşir. Erişim noktası sağlayıcısının ağı da, sabit ağı kalan kısmına tek bir mantıksal grup düğümü olarak birleşir. Erişim noktası sağlayıcısının, sabit ağı geri kalan kısmına sadece özetlenmiş ulaşılabilirlik bilgisi tanıtması tavsiye edilir. Özet ulaşılabilirlik bilgisinin içinde, erişim noktasının altyapısının adresleri ve sağlayıcının hierarchyesine bağlanmış mobil ağların tüm adresleri bulunmalıdır.



Şekil 5.9 Bir ağı yapısı

ATM ağındaki adres ön ekleri ve düğüm sayısı açısından ölçeklenebilirliği sağlamak için bu yapı ve özetleme kurallarına uyulmalıdır. Ulaşılabilirlik bilgisinin yetersiz ve hatalı özetlenmesi, PNNI topoloji veritabanının büyümesine ve ağıdaki düğümlerin başarısızlığına sebep olabilir.

### 5.2.6 Yeniden Yönlendirme

Bu bölümde, ağı tarafından varolan sanal anahtar bağlantısını (SVC), sanal anahtar güzergahını (SVP), soft kalıcı sanal bağlantıyı (Soft PVC) veya soft kalıcı sanal güzergahı (Soft PVP), otomatik olarak kullanıcı etkileşimi gerekmeden, yeniden

yönlendirilmesi açıklanmaktadır. Bu protokol, PNNI v1.0 işaretleşme protokolüne yeni bir işlevin eklenmesiyle oluşturulmuştur[35].

Yeniden yönlendirmeyi desteklemek için bazı alternatif yaklaşımları olan bir yeniden yönlendirme mekanizması geliştirilmiştir. Çeşitli gereksinimler için, farklı kullanıcı grupları için farklı alternatifler arzu edilebilir. Bu yüzden, yeniden yönlendirme protokolü çeşitli yeniden yönlendirme mekanizmalarını destekleyebilecek şekilde tasarlanmalıdır.

Burada, noktadan-noktaya bağlantılar için kenardan-kenara yeniden yönlendirme mekanizmasını anlatılmaktadır. Bu mekanizmada, kaynak düğüm (mesela DTL yaratıcısı) ve hedef düğüm (mesela DTL sonlandırıcısı) yeniden yönlendirme işleminin kontrolüne katılmıştır.

Yeniden yönlendirme mekanizmasında kullanılan çeşitli terimler aşağıda açıklanmıştır:

**Bağlantı Segmenti:** Bir bağlantının bir bölümünün veya bir bağlantının tamamının, bir ya da daha fazla PNNI ağları üzerinden, bağlantıdaki ara düğümlerle bütünleşmesi.

**Yeniden Yönlendirme Bağlantı Segmenti:** Eski bağlantı segmentini değiştirmek için kullanılan bir bağlantı segmenti.

**Kaynak Düğüm:** Kaynak terminale bağlanmış bir düğüm veya bu bağlantının oluşturduğu bir uç sistem.

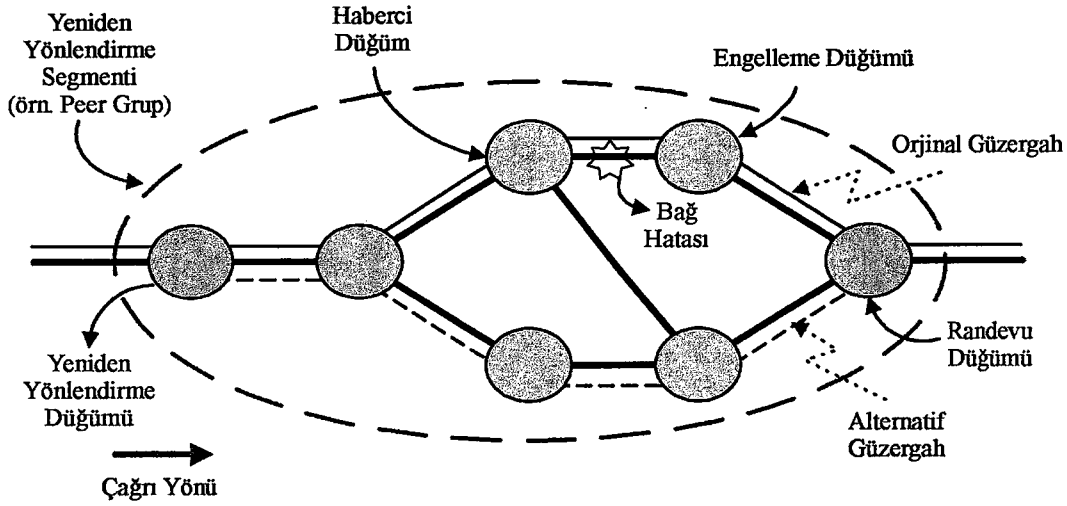
**Hedef Düğüm:** Hedef terminale bağlanmış bir düğüm veya bağlantının sona erdiği bir uç sistem.

**Yeniden Yönlendirme Düğümü:** Daha önceden belirlenmiş olan bir sonlandırma düğümüne alternatif bağlantı segmenti oluşturma işleminden sorumlu olan düğüm.

**Randevu Düğümü:** Alternatif bağlantı segmenti için yeniden yönlendirme talebini sonlandırmakla sorumlu olan düğüm.

**Haberci Düğüm:** Bir bağlantıdaki hata noktasından hemen önce gelen, yeniden yönlendirme düğümü doğrultusunda bir bağlantı segmenti oluşturmakla ve eğer

yeniden yönlendirme gerekiyorsa, yeniden yönlendirme düğümündeki yeniden yönlendirme prosedürlerini tetiklemekle sorumlu olan düğümdür.



Şekil 5.10 Yeniden Yönlendirme Örneği

**Engelleme Düğümü:** Bir bağlantıdaki hata noktasını hemen takip eden, randevu düğümü doğrultusunda bir bağlantı segmenti oluşturmakla ve eğer yeniden yönlendirme gerekiyorsa randevu düğümündeki yeniden yönlendirme prosedürlerini tetiklemekle sorumlu olan düğümdür.

**Yeniden Yönlendirilebilirlik:** Talep edilen yeniden yönlendirme servisini (mesela bağlantı onarımı) ve kullanıcı izinli ağ yeniden yönlendirme seçeneğini (yönetimsel yeniden yönlendirme) esas alarak yeniden yönlendirme prosedürlerini oluşturma kabiliyetidir.

**Kenardan-Kenara Yeniden Yönlendirme:** PNNI ağındaki iç kenarlarda bulunan anahtardan başlayan ve dış kenarda bulunan anahtarda sona eren bir bağlantı segmentini değiştirmek için tasarlanmış bir yeniden yönlendirme mekanizmasıdır.

**Farklı Güzergaha Yeniden Yönlendirme:** Daha önceden belirlenmiş farklı bir bağlantı segmentine, bağlantı segmentini göç ettirerek, bir bağlantı segmentini değiştirmek için tasarlanmış bir yeniden yönlendirme tipidir.

**Alternatif Hedefe Yeniden Yönlendirme:** Seçilen düğümden hedef düğüme kılavuzluk eden bir bağlantı segmentinin yine aynı düğümden alternatif bir hedefe

gitmesi için başka bir bağlantı segmentiyle değiştirilmesi amacıyla tasarlanmış bir yeniden yönlendirme tipidir.

**Bağlantı Onarımı:** Bağlantıdaki bir veya daha çok noktadaki kaynak hatasına bağlı olarak bağlantının onarılmasını talep eden bir yeniden yönlendirme sebebidir.

**QoS Yenilenmesi:** Daha önceden kabul edilen bir QoS garantisinin artık sağlanmaması durumu, QoS yenilenmesini gerektiren bir tekrar yönlendirme sebebidir.

**Güzergah Optimizasyonu:** Daha önceden belirlenmiş yönlendirme amaçlarını temel alarak bir bağlantının yeniden optimizasyonunu talep eden bir yeniden yönlendirme sebebidir.

**Öncelik Kontrolü:** Bir bağlantının tekrar kurulmasını diğer bağlantıya verilen öncelik hakkı nedeniyle gerektiren yeniden yönlendirme sebebidir.

**Yönetimsel Yeniden Ayırma:** Daha önceden belirlenmiş bir yönetimsel amacın gerçekleştirilebilmesi için kurulmuş bağlantılar arasındaki ağ kaynaklarının yeniden ayrılmasını gerektiren yeniden yönlendirme sebebidir.

**Kenardan-Kenara Yeniden Yönlendirme,** PNNI ağındaki iç kenarlarda bulunan anahtardan başlayan ve dış kenarda bulunan anahtarda sona eren bir bağlantı segmentini değiştirmek için tasarlanmış bir yeniden yönlendirme mekanizmasıdır. Kenardan-kenara yeniden yönlendirme SVC, SVP, Soft PVC ve Soft PVP için, PNNI ağındaki bir kaynak düğümden başlayarak (mesela DTL yaratıcısı) hedef düğümde sonlanan (mesela DTL sonlandırıcısı) bir bağlantı segmentini değiştirmek için tasarlanmıştır.

Kenardan kenara yeniden yönlendirmeyi gerçekleştirmek için desteklenen iki yaklaşım vardır; “Break Before Make” ve “Make Before Break”.

“Break Before Make Rerouting”, bağlantının alternatif bağlantı segmentinin kurulmasından önce oluşturulduğu yeniden yönlendirme prosedürüdür. “Hard Reroute” olarak da bilinmektedir.

“Make Before Break Rerouting” ise, bağlantının sadece alternatif bağlantı segmentinin kurulmasından sonra oluşturulduğu yeniden yönlendirme prosedürüdür. “Soft Reroute” olarak da bilinmektedir.

“Break Before Make” prosedürlerinin tasarımı bağlantı onarımı veya öncelik kontrolü özellikleri için kullanılabilir. Güzergah optimizasyonu ve yönetsel yeniden yönlendirme gibi diğer kullanışlı yeniden yönlendirme özellikleri için “Make Before Break” prosedürleri gereklidir.

Soft Reroute halinde, sadece yeniden yönlendirme düğümü ve randevu düğümü arasındaki yeniden yönlendirme işleminde kullanılan prosedürler belirtilir.

Kenardan kenara yeniden yönlendirme için, DTL yaratıcısı (yeniden yönlendirme düğümü) ve DTL sonlandırıcısı (randevu düğümü), yeniden yönlendirme operasyonunun bağlantı kontrolüne iştirak eder. “Break Before Make” ve “Make Before Break” yeniden yönlendirme tipleri arasındaki olası çakışmanın kolayca önüne geçilebilmesi için her bağlantı segmentinin sonuna protokoller arası koordinasyonu sağlayacak bir yeniden yönlendirme durum makinası tasarlanmıştır. Yeniden yönlendirme prosedürlerinin basitleştirilmesi için, bir seferde sadece bir bağlantının bir anahtarda gerçekleştirilebileceği, yalnızca bir yeniden yönlendirme operasyonuna izin veren bir anlaşma yapılmıştır. Fakat Soft Reroute ve Hard Reroute’un farklı tetikleme yapılarına bağlı olarak, Soft Reroute operasyonunun Hard Reroute operasyonu tarafından kesilmesi muhtemeldir. Bu durumda Hard Reroute, Soft Reroute’un yerini alır ve Hard Reroute ile operasyon devam eder. Sonuç olarak yeniden yönlendirme makinesinin tasarımı bu kabulü esas alır.

Yeniden yönlendirme eklentisi, halen tasarım aşamasındadır. ATM Forum çıkaracağı PNNI v2.0’a bu eklentinin son halini koyacaktır. PNNI v2.0, PNNI v1.0’a yapılan eklentiler ve halen yeniden yönlendirme gibi taslak aşamasında olan çalışmaların birleşiminden oluşacaktır.

## 6 SONUÇLAR

PNNI protokolü ölçeklenebilir, hizmet kalitesi gözeten ATM yönlendirmesi ve anahtarlar arası sanal bağlantıları destekleyen mekanizmaları bünyesinde bulundurur. PNNI bunu gerçekleştirmek için işaretleşme ve yönlendirme protokollerinden faydalanır.

PNNI işaretleşmesi, UNI işaretleşmesinin NNI bağları üzerinde kullanılabilmek üzere geliştirilmiş bir halidir, ek bilgi elemanlarını (DTL gibi) ve ek mekanizmaları (Cranckback gibi) içerir.

PNNI protokolünün ikinci bileşeni ise bir yönlendirme protokolüdür ve işaretleşme talebinin ATM ağı içerisinde yönlendirilmesi için kullanılır. Bir ATM ağında işaretleşme talebinin yönlendirilmesi, ATM'in bağlantı tabanlı yapısına rağmen, diğer ağ katmanı protokollerindeki (IP gibi) bağlantısız paket yönlendirmesine benzerlik göstermektedir. Bunun sebebi ise bağlantı kurulmasından önce, işaretleşme talebinin takip edeceği herhangi bir bağlantının bulunmamasıdır.

PNNI yönlendirme protokolü büyük ağları ölçeklendirmeye izin vermesi ve QoS ile yönlendirmeyi desteklemesi nedeniyle varolan yönlendirme protokollerinden daha karmaşıktır.

QoS garantisinin sağlanması için düğümler bağlantı kabul denetimi adlı yerel bir fonksiyon (CAC) kullanılırlar. CAC fonksiyonunun nasıl gerçekleştirileceği bir standarda bağlı değildir, yerel politikaların belirlenmesi kullanıcıya bırakılmıştır.

PNNI yönlendirme protokolü bağ durumu yönlendirme algoritması temel alınarak geliştirilmiştir. Bu algoritmanın geniş çaplı ağları desteklemekteki dezavantajı hiyerarşik yönlendirme mekanizmaları eklenerek aşılmıştır. Bu kavram dahilinde, ağdaki düğümler ve bağlar hiyerarşik olarak organize edilmişlerdir. Yönlendirme için gerekli ulaşılabilirlik ve QoS bilgisi PNNI topoloji durum elemanları (PTSE) halinde kodlanarak ağ içine dağıtılır. Bu yapıyla OSPF gibi güncel bağ durumu yönlendirme protokollerine benzerdir. QoS için sadece basit bir desteğe sahip olan

bu protokollerden farklı olarak, PNNI protokolü, düğümler tarafından iletilen büyük miktarlardaki bağ ve düğüm durum parametrelerini de destekler.

Ağ üzerinden yapılacak bir yönlendirme bağlantısı için iki yöntem mevcuttur. Kaynaktan yönlendirme ve adım adım yönlendirme. Adım adım yönlendirme protokolüyle doğru QoS tabanlı yönlendirme yapmak zordur çünkü her düğümün yerel CAC gerçekleştirmesi ve bir sonraki adımı bulmak için bütün ağ içinde QoS'yi değerlendirmelidir. Fakat, kaynaktan yönlendirmeli protokolde, sadece ilk düğüm, güzergahı belirler.

Ancak bu tanım en ideal durum için geçerlidir. Uygulamada, bir düğüm tarafından belirlenen kaynaktan yönlendirmeli güzergah sadece iyi bir tahmin olabilir. Bunun nedeni uygulamadaki ağlarda herhangi bir düğümün, ağın gerçek durumu hakkında sadece mükemmel olmayan bir tahmin yapabilmesidir; çünkü PTSP taşkın yöntemindeki periyod uzunlukları ve gecikmeleri bunu engellemektedir. Ulaşılabilirlik bilgisinin hiyerarşik özetlenmesi, bağ parametrelerinin de birleştirilmesini gerektirir. Birleştirme kayıplı bir işlemdir ve yanlışlıklara yol açmaktadır. Buna ek olarak CAC yerel bir sorundur. Uygulamada herhangi bir düğüm tarafından gerçekleştirilen CAC algoritması hem sistem bağımlıdır hem de üretici farklılaşmalarına açıktır.

PNNI protokolü bu problemleri genel bir bağlantı kabul denetimi (GCAC) algoritması tanımlayarak kontrol altına alır. Bu, her düğümün, başka bir düğümün beklenen CAC davranışını o düğüme ait toplanabilir bağ metriklerini kullanarak hesaplamak için ve yeni bağlantı talebinin talep ettiği QoS'u sağlayıp sağlamadığını tahmin etmek için kullanılan standart bir fonksiyondur. GCAC'ın farklı tipleri, talep edilen QoS garantisinin tipine bağlı olarak ve kullanılabilir bağ metriklerinin tiplerine bağlı olarak, farklı yanılma paylarına sahiptir.

Bunlara ek olarak, güzergahtaki her düğüm, yönlendirme talebi üzerinde Kendi CAC'ını gerçekleştirir çünkü kaynak düğümündeki GCAC için kullanılan PTSP içindeki düğümün, o anki durumu son tanıtım durumundan itibaren değişiklik göstermiş olabilir. Aynı zamanda kendi CAC algoritması, GCAC'dan daha doğru olabilir. Dolayısıyla talebin reddedilme olasılığı vardır. Bu durum çok hiyerarşik seviyeli geniş ağlarda, QoS bilgisi doğru olarak bildirilemediği için, gerçekleşebilir. Bu tür durumlardaki aşırı bağlantı hatalarının veya tekrar denemelerin önüne geçmek için, PNNI protokolü crankback olayını destekler. Halen Cranckback algoritmasının

iyileştirmesi yönünde çeşitli öz kaynak kotaları kullanan önceden tahmin algoritmaları üzerine çalışılmaktadır.

PNNI yönlendirmede kullanılan algoritmalar (Dijkstra; en kısa yol algoritması gibi) tek bir birikimli metrik kullanırlar. PNNI birçok kompleks bağ parametresi kullandığı için, güzergah seçimi sıkça kabul edilebilir bir güzergah oluşturamaz. Bu gibi durumlarda, gelişmiş algoritmalar kullanılabilir. Bu algoritmalar, bazı kısıtları (örneğin gecikme gibi) gevşeten ve güzergahları arzu edilen minimal özellikler kümesini karşılayacak şekilde yeniden hesaplayan algoritmalarlardır. Genel olarak güzergah seçimi, CAC gibi, üretici farklılıklarına açık olan bir alandır.

QoS desteği sağlamanın yanında, ATM Forum, PNNI protokolüne evrensel ölçeklenebilirlik hedefini koymuştur. PNNI protokolü birkaç anahtarlık küçük ağlara hizmet etmenin yanında, gelecekteki milyonlarca anahtardan oluşacak olan küresel ATM internet beraberliğini destekleyebilecek şekilde tasarlanmıştır. İnternet günümüzde IGRP, OSPF, BGP ve IDRIP gibi birçok farklı tipteki yönlendirme protokollerini desteklemektedir, oysaki ATM Forum PNNI ile bir ağ içerisindeki bütün seviyelerde işleyebilecek tek bir protokol geliştirmiştir.

Sonuç olarak, ATM Forum PNNI v1.0'ı çıkardığından itibaren gerek değişen piyasa koşulları gerekse teknolojik gereksinimler doğrultusunda PNNI üzerinde çeşitli eklentiler ve düzeltmeler yapmıştır. PNNI protokolünün, başarımını etkileyen tasarım kararları hakkında çalışmalar halen sürmektedir. Bu çalışmalar arasında en önemlileri ATM protokolünün mevcut ağ protokolleriyle, popüleritesinden ötürü özellikle IP protokolüyle, olan iletişimini ve bütünleşmesini kapsamaktadır. ATM Forum bu konuda başarılı çalışmalarını sürdürürse, yakın gelecekte PNNI, ölçeklenebilirlik ve hizmet kalitesi garantisi gibi önemli avantajlarıyla sadece ATM ağları için değil, mevcut diğer altyapılar içinde tercih edilen bir çözüm olacaktır.

## KAYNAKLAR

- [1] McDysan, E.D. ve Sphon, D.L., 1995. ATM Theory and Application, McGraw Hill.
- [2] Händel, R., Huber, M. ve Schröder, S., 1994. ATM Networks, Ed. 2, Addison Wesley.
- [3] Cherukuri, R. ve Onvural, Raif O., 1997. Signaling in ATM Networks, Artech House.
- [4] Guizani, M. ve Rayas, A., 1999. Designing ATM Switching Networks, McGraw Hill.
- [5] ATM Forum Technical Committee, 1996. Private Network-Network Interface Specification Version 1.0.
- [6] Streenstrup, M.E., 1995. Routing in Communication Networks, Prentice Hall.
- [7] Alles, A., 1995. ATM Internetworking, Cisco Systems.
- [8] ATM Forum Technical Committee, July 1996. ATM User-Network Interface (UNI) Signaling Specification Version 4.0.
- [9] Felstaine, E., Cohen, R. ve Hadar, O., 1998. Crankback Prediction in Hierarchical ATM Networks, *Technion Computer Science Department Technical Report*, LCCN9807.
- [10] ATM Forum Technical Committee, March 1999. Traffic Management Specification Version 4.1.
- [11] Singh, G., 1996. Leader Election in the Presence of Link Failures, *IEEE Transactions on Parallel and Distributed Systems*, 7, (3), 231-236.
- [12] Awerbuch, B., Du, Y., Khan, B. ve Shavitt, Y., 1998. Routing Through Networks with Hierarchical Topology Aggregation, *DIMACS Technical Report 98-16*, Department of Computer Science John Hopkins Baltimore.

- [13] **Huitema, C.**, 1995. Routing in The Internet, Prentice Hall.
- [14] **Felstaine, E. ve Cohen, R.**, 1999. On The Distribution of Routing Computation in Hierarchical ATM Networks, *IEEE Transactions on Networking*, **7**, (6), 906-912.
- [15] **Black, U.**, 2000. IP Routing Protocols RIP, OSPF, BGP, PNNI, and Cisco Routing Protocols, Prentice Hall.
- [16] **Lee, Whay C.**, 1995. Topology Aggregation for Hierarchical Routing in ATM Networks, *Computer Communications*, **25**, (2), 82-92.
- [17] **Lee, Whay C.**, 1994. A Minimal Spanning Connectivity Method for Link State Aggregation, *ATM Forum*, 94-0673.
- [18] **Awerbuch, B., Du, Y. ve Shavitt, Y.**, 2000. The Effect of Network Hierarchy Structure on Performance of ATM Hierarchical Routing, *Computer Communications*, **23**, 980-986.
- [19] **Giroux, N. ve Ganti, S.**, 1999. Quality of Service in ATM Networks, Prentice Hall.
- [20] **Black, U.**, 2000. QOS In Wide Area Networks, Prentice Hall.
- [21] **Iwata, A., Izmailov, R., Suzuki, H., Sengupta, B.**, 1997. PNNI Routing Algorithms for Multimedia ATM Internet, *Nec Research & Development*, **38**, (1).
- [22] **Jocher, P. ve Frings, J.**, 1998. Planning Aspects of ATM Networks Using The PNNI Routing Protocol, *Proceedings NOC'98 European Conference On Networks & Optical Communications*, England.
- [23] **Basturk, E. ve Stirpe, P.**, 1997. A Hybrid Spanning Tree Algorithm for Efficient Topology Distribution in PNNI, *IBM Research Report, RC20922*, IBM Research Division.
- [24] **ATM Networking Group Zurich Research Lab**, 1997. IBM PNNI Control Point Switched Network Series, IBM.
- [25] **Chen, X. ve Tantiprasut, Duke T.**, 1997. Evolution of ATM Internetworking, *Bell Labs Technical Journal*, **Spring 97**.
- [26] **ATM Forum Technical Committee**, 1996. An Overview of PNNI Augmented Routing.

- [27] **ATM Forum Technical Committee**, 1999. PNNI Augmented Routing (PAR) Version 1.0.
- [28] **ATM Forum Technical Committee**, 1996. Issues and Approaches for Integrated PNNI.
- [29] **ATM Forum Technical Committee**, 1997. Integrated PNNI (I-PNNI) v1.0 Specification.
- [30] **ATM Forum Technical Committee**, 1999. PNNI Addendum for Generic Application Transport Version 1.0.
- [31] **ATM Forum Technical Committee**, 1999. PNNI Version 1.0 Security Signaling Addendum.
- [32] **ATM Forum Technical Committee**, 1999. ATM Security Specification Version 1.0.
- [33] **ATM Forum Technical Committee**, 2000. PNNI Addendum for Path and Connection Trace, Version 1.0.
- [34] **ATM Forum Technical Committee**, 1999. PNNI Addendum for Mobility Extensions Version 1.0.
- [35] **ATM Forum Technical Committee**, 1998. Proposed Baseline Text for Edge-to-Edge Rerouting for Point-to-Point Connections.

## ÖZGEÇMİŞ

Cumhur Okan Özoğul, 9 Ekim 1976'da İstanbul'da doğmuştur. Orta ve lise öğrenimini, Saint-Benoit Fransız Lisesi'nde tamamlamıştır.

1994-1998 yılları arasında Galatasaray Üniversitesi Bilgisayar Mühendisliği Bölümünde eğitimini tamamlamıştır.

1998 yılında, İstanbul Teknik Üniversitesi'nde Kontrol ve Bilgisayar Mühendisliği Yüksek Lisans Programına başlamıştır. 1999 yılı Temmuz ayından beri, Galatasaray Üniversitesi'nde Araştırma Görevlisi olarak çalışmaktadır.

