

향상된 Star Topology Aggregation 기법 Improved Star Topology Aggregation Scheme

김남희*, 조해성

군산대학교 컴퓨터정보과학과*, 건양대학교 전자정보
공학과

Kim Nam-Hee*, Cho Hae-Seong

Computer Information Science Dept.
Kunsan National University*
Electronic & Information Engineering
Dept. Konyang university

요약

본 논문에서는 대역폭과 지연 파라미터를 참조하여 PG내의 토폴로지 정보를 요약하는 기법으로 라인 세그먼트를 이용하여 경계노드 사이의 다중 경로 정보를 요약하였다. 이를 위해 토폴로지 요약 정보를 줄이고 다중링크 요약에 유연성을 부여하기 위하여 대역폭과 지연의 한 쌍으로 구성된 두 개의 라인 세그먼트를 이용하는 수정된 라인 세그먼트 기법을 제안하였다. 시뮬레이션 분석결과 제안된 star TA 기법이 기존의 기법보다 성능이 향상됨을 알 수 있었다.

Abstract

In this paper, we aggregate multi-links information between boundary nodes using the line segment scheme that aggregates topology information within PG referring bandwidth and delay parameter. To do this, we propose a modified line segment algorithm using two line segment method that represents two points which consist of delay-bandwidth pair to reduce topology information and provide a flexibility to the multiple-links aggregation. And we apply it to current star topology aggregation. Through the simulation result analysis, the proposed star topology aggregation scheme presents the better performance than existing scheme.

I. 서론

PNNI 규격은 복잡한 구조를 가지고 있어 공중망에 적용하기는 현실적인 어려움이 있지만, 망의 동적인 변화에 적응성이 뛰어나고, 다양한 QoS(Quality of Service) 제공 능력을 가지고 있다[1,2]. 근래의 PNNI 규격은 대부분의 상용 사실 ATM 스위치에서 구현되고 있으며, 공중망에의 적용 가능성과 함께 B-ISUP 공중망간의 효율적인 상호 연동을 위한 많은 연구가 진행되고 있다. Topology aggregation은

peer group 내부 토폴로지 정보를 하나의 가상 노드(LGN : Logical Group Node)로 요약하는 것이다. 토폴로지 요약정보는 LGN 노드 도달 정보, 다른 LGN과의 가상 논리 링크 상태 정보, LGN 내부 구조 및 상태 정보가 포함된다. LGN 노드 도달 정보는 ATM 주소의 prefix를 이용하여 요약된다. 다른 LGN과의 가상 논리 링크 상태 정보는 PNNI망의 계층적인 구조에 따라 결정된다. 그러나, LGN 내부 구조 및 상태 정보는 토폴로지 요약 기법에 따라 정보

량 및 정확성에 많은 차이가 나기 때문에 초기의 연구는 가상적인 내부 구조를 중심으로 이루어졌다 [3,4]. 본 논문에서는 peer group 내의 토폴로지 정보를 요약하는 기법으로 대역폭과 지연 파라미터를 중심으로 라인 세그먼트를 이용하여 경계노드 사이의 다중 경로 정보를 요약하였다. 라인 세그먼트 파라미터 값을 결정하기 위해서는 모든 경로를 찾아야 한다. 그러나 모든 경로를 찾기 위해서는 많은 시간이 필요하며 이렇게 찾아진 경로들 중에서 일부만이 라인 세그먼트 파라미터 값을 결정하기 위해 사용된다. 따라서 모든 경로를 찾는 대신 홉 카운트 기반의 깊이 우선방식을 사용하여 효율적으로 다중 경로를 탐색할 수 있도록 하였다[5]. 즉, 정보량을 줄이고 다중 링크 요약에 유연성을 부여하기 위해 두 개의 라인 세그먼트를 이용하여 기존의 기법보다 향상된 효율적인 star TA(Topology Aggregation) 기법을 제안하였다. 하나의 라인 세그먼트를 표현하기 위해서는 양 끝점의 대역폭과 지연 파라미터가 필요하므로 네 개의 파라미터 값이 필요하게 된다. 본 논문에서는 하나의 라인 세그먼트를 표현하기 위해 필요한 네 개의 파라미터 값을 가지고 두 개의 라인세그먼트를 표현하여 다중 링크 요약에 유연성을 부여하였다. 또한, 면적 개념을 도입하여 다중 QoS 파라미터로 하나의 대표값을 사용하여 star TA 기법에 적용하여 계산의 복잡도를 감소시켰다. 그리고 요약 기법의 성능 분석을 컴퓨터 시뮬레이션을 통하여 기존의 알고리즘과 호 성공율, 접근시간 및 크랭크 백 울에 대하여 비교·분석하였다.

2. 제안된 Star Topology Aggregation

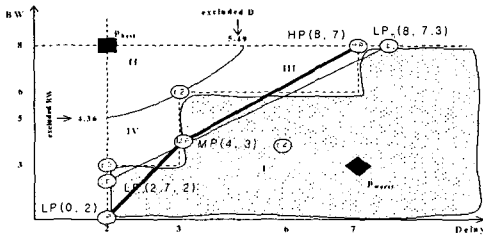
2.1 Modified line segment

라인 세그먼트 기법은 LP_1 와 LP_2 를 결정하여 QoS 제공 파라미터 영역을 설정하는 방식으로 네 개의 파

라미터를 가지고 다중 링크 QoS를 표현할 수 있으므로 다중 링크를 효과적으로 요약할 수 있는 방법이라 할 수 있다. 전파 지연과 제공 대역폭을 이용한 요약 기법이지만, ATM 스위치에서 Fair Queueing 스케줄링 알고리즘을 이용하면 다른 QoS 파라미터를 구할 수 있으므로, 이를 이용하여 경로 선택 알고리즘에 적용하면 다중 QoS 파라미터를 이용한 경로 선택이 가능하다. 제안된 방식은 [그림 1]에서와 같이 세 개의 점 LP, MP, HP를 가지고 두 개의 라인 세그먼트를 이용하여 다중 링크 QoS를 표현하는 방식이다. 세 개의 점을 표현하기 위해서는 6개의 파라미터가 필요하나 파라미터의 수를 줄이기 위해 한 개의 대역폭 파라미터와 세 개의 지연 파라미터를 가지고 요약하였다. 즉, [그림 1]의 LP(0, ld), MP(Max BW/2, md), 그리고 HP(Max BW, hd) 세 개의 파라미터를 사용해 표현함으로써 하나의 라인 세그먼트와 같은 양의 정보를 가지고 두 개의 라인 세그먼트를 표현할 수 있으므로 다중 링크 요약에 있어서 유연성을 제공할 수 있게 된다. [그림 1]에 제안된 modified line segment는 가장 간단하게 구할 수 있는 라인 세그먼트 방식을 나타낸 것으로 이는 라인 세그먼트를 결정하는 방식에 다양하게 적용시킬 수 있다. 본 논문에서는 제안된 라인 세그먼트 방식을 star TA 기법에 적용하여 기존의 star TA 보다 효율적인 TA기법을 제안하였다.

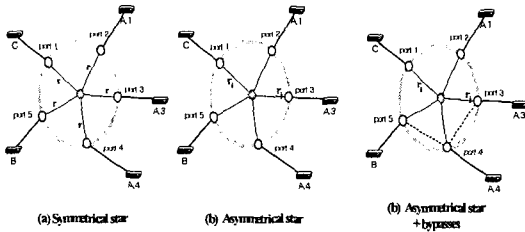
2.2 제안된 star topology aggregation 기법

Star 구조는 논리 링크 상태 파라미터 결정 방법에 따라 [그림 2]에 보여지듯이 symmetric star, asymmetric star, bidirectional asymmetric star 구조가 있다. Star 구조는 LGN 내부 구조에 가상 노드를 추가하는데 이를 nucleus node라 한다. 또한 포트와 nucleus 노드 사이의 논리 링크를 spoke link라 한다.



▶▶ 그림 1. 제안된 라인 세그먼트에서의 재공가능영역

Symmetric star 구조의 spoke link 상태 파라미터는 simple node 구조의 상태 파라미터와 같은 방법으로 구하여 반으로 나눈 값이 되며 이 구조는 많은 압축 효과를 볼 수 있지만, full mesh 구조와 많은 차이가 난다. 이를 보완하기 위해 bypass link를 추가하면 복합 구조 TA 기법이 된다.



▶▶ 그림 2. Star 구조

이들을 차이가 많이 나는 순서로 정렬하고, 임계치를 이용하거나 일정한 수로 bypass link를 결정한다. Bypass link가 결정되면 링크의 QoS 파라미터를 full mesh 구조와 같은 값으로 하여 star 구조에 포함시킨다. 이제 bypass link가 추가되었기 때문에 spoke link의 QoS 파라미터를 다시 계산한다. 이 과정은 full mesh 구조에서 bypass link와 같은 포트 사이의 논리 링크를 제거하고 spoke link QoS 파라미터를 다시 구하면 된다. PG의 토폴로지 정보를 요약하기 위해 LGN 내부 구조를 symmetric star 구조에 bypass link를 추가한 복합 구조를 이용하면 star 구조의 특성을 살릴 수 있고, 작은 정보량으로 보다 정확한 요약 정보를 얻을 수 있다. Symmetric star 구조는 포트 사이의 QoS 파라미터 값이 비슷한 대칭

구조에 적합한 구조라 할 수 있는데, QoS 파라미터는 동적으로 변하므로 이에 적용하기 위해 asymmetric star 구조를 이용한다. 각 포트의 spoke link의 대역폭 파라미터는 full mesh 구조의 각 포트에서 나가는 논리 링크의 가장 작은 대역폭 파라미터가 된다. 지연 파라미터는 모든 spoke link의 지연 파라미터를 0으로 초기화 하고, 제약 조건을 만족시키기 위해 더해 주어야 하는 값들 중에서 가장 작은 값을 찾아서, 반으로 나누어 spoke link에 더해준다. 더 이상 더해줄 값이 없을 때까지 즉, 모든 제약 조건을 만족할 때까지 위의 과정을 반복한다. Asymmetric star 구조는 symmetric star 구조에 비해 정보량은 다소 많으나, QoS 파라미터의 동적 변화에 적용할 수 있는 기법이다. Asymmetric star 구조는 spoke link의 방향성이 고려되지 않았으나 spoke link의 양 방향의 링크 상태 QoS 파라미터를 다르게 구한 것이 bidirectional asymmetric star 구조이다. Bidirectional asymmetric star 구조는 정보량은 다른 star 구조에 비해서 많으나 full mesh 구조와 거의 같은 수준으로 요약할 수 있다. Full mesh 구조의 정보량에 비해 0.5 / (Np-1) 정도이며 bypass link를 추가하더라도 1.5 / (Np-1) 정도이므로 포트 수가 커질수록 많은 압축 효과가 있다.

Bidirectional asymmetric star 구조에서 spoke link의 QoS 상태 파라미터 값을 구하기 위해 제안한 라인 세그먼트의 연산들은 다음과 같다.

```

+ 연산 (addLineSG = lineSG1 + lineSG2) (그림 3(a))
sub function addLineSParam (lineSG1, lineSG2)
begin
    HighDelay, MidDelay : delay parameter
    addLineSG : result of add line segments
    addLineSG.maxBW = min (lineSG1.maxBW, lineSG2.maxBW)
    if (lineSG1.maxBW > lineSG2.maxBW) then
        MidDelay = getDelay (lineSG1.lowLine, lineSG2.maxBW/2)
        HighDelay = getDelay (lineSG1.highLine, lineSG2.maxBW)
    addLineSG.lowDelay = lineSG2.lowDelay+lineSG1.lowDelay
    addLineSG.midDelay = lineSG2.midDelay + MidDelay

```

```

    addLineSG.highDelay = lineSG2.highDelay +
HighDelay
    else
        MidDelay = getDelay (lineSG2.lowLine,
lineSG1.maxBW/2)
        HighDelay = getDelay (lineSG2.highLine,
lineSG1.maxBW)
        addLineSG.lowDelay =
lineSG1.lowDelay+lineSG2.lowDelay
        addLineSG.midDelay = lineSG1.midDelay +
MidDelay
        addLineSG.highDelay = lineSG1.highDelay +
HighDelay
    end if
end function

```

```

- 연산 (subLSG = lineSG1 - lineSG2) (그림 3(b))
sub function subLineSParam (lineSG1, lineSG2)
begin
    HighDelay, MidDelay : delay parameter
    subLSG : result of minus line segments
    subLSG.maxBW = min (lineSG1.maxBW,
lineSG2.maxBW)
    if (lineSG1.maxBW > lineSG2.maxBW) then
        MidDelay = getDelay (lineSG1.lowLine,
lineSG2.maxBW/2)
        HighDelay = getDelay (lineSG1.highLine,
lineSG2.maxBW)
        subLSG.lowDelay = lineSG2.lowDelay -
lineSG1.lowDelay
        subLSG.midDelay = lineSG2.midDelay -
MidDelay
        subLSG.highDelay = lineSG2.highDelay -
HighDelay
    else
        MidDelay = getDelay (lineSG2.lowLine,
lineSG1.maxBW/2)
        HighDelay = getDelay (lineSG2.highLine,
lineSG1.maxBW)
        subLSG.lowDelay = lineSG1.lowDelay -
lineSG2.lowDelay
        subLSG.midDelay = lineSG1.midDelay -
MidDelay
        subLSG.highDelay = lineSG1.highDelay -
HighDelay
    end if
end function

```

```

지연 값이 주어졌을 때의 면적 (그림 3(c))
a : constant(1 ~ 2)
totalBW : total bandwidth of physical links
linkSG : Line segment of link
nDelay : The normalize value of delay
MaxDelay = max (link.highDelay in full mesh) × a
nBW : The normalize value of bandwidth
nDelay = (MaxDelay - linkSG.highDelay) / MaxDelay
nBW = linkSG.maxBW / totalBW
AREA = nDelay × nBW
nDelay = (linkSG.highDelay - linkSG.midDelay) /
MaxDelay
AREA = AREA + (3 × nDelay × nBW / 4)
nDelay = (linkSG.midDelay - linkSG.lowDelay) /
MaxDelay
AREA = AREA + (nDelay × nBW / 4)

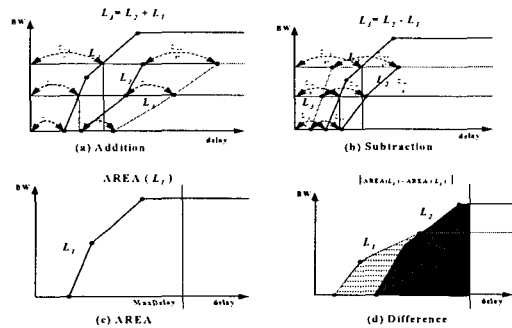
```

```

두 라인 세그먼트의 차이도 (그림 3(d))
lineSG1, lineSG2 : line segment
Difference = | AREA(lineSG1) - AREA(lineSG2) |

```

Bidirectional asymmetric star 구조의 spoke link 는 양방향이므로 포트에서 nucleus node로의 방향과 반대 방향의 라인 세그먼트 파라미터를 구해야 한다. Nucleus node로 들어가는 방향을 inSpoke link, 반대 방향을 outSpoke link라고 하면. 먼저 full mesh 구조의 링크 상태 파라미터들을 이용하여 inSpoke link의 라인 세그먼트 파라미터를 구하고, full mesh 와의 차이를 이용하여 outSpoke link의 라인 세그먼트 파라미터를 구한다. 각 포트에서 inSpoke link의 라인 세그먼트 파라미터는 full mesh 구조의 같은 포트에서 나가는 논리 링크 상태 파라미터들에서 가장 좋은 파라미터 값을 가지게 된다.



▶▶ 그림 3. 라인 세그먼트 연산

Full mesh 구조의 포트 i에서 포트 j로의 논리 링크는 star 구조의 inSpoke link i와 outSpoke link j의 합을 포함해야 한다. 그러므로 outSpoke link j의 라인 세그먼트 파라미터는 full mesh link ij와 inSpoke link i의 차와 같거나 커야 한다. 그런데 full mesh 구조에서 포트 j로 향하는 링크는 여러 개가 있으므로 이들의 평균값을 구하여 outSpoke link j의 라인 세그먼트 파라미터를 구한다.

inSpoke link와 outSpoke link의 라인 세그먼트 파라미터를 구하는 알고리즘은 다음과 같이 나타낼 수 있다.

```

For each inSpoke links
  outlink : At the same port in full mesh
  lowDelay = min (outlink.lowDelays in full mesh)
  midDelay = min (outlink.midDelays in full
mesh)
  highDelay = min (outlink.highDelays in full
mesh)
  maxBW = max (outlink.maxBW in full mesh)
END For
For each outSpoke_j links
  FMlink_ij : The link from port i to port j in full
mesh
  For each FMlink_ij port i
    DParam = FMlink_ij - inSopke_i
    listDParams.addParam (DParam)
  END For
  lowDelay = average (DParam.lowDelays in
listDParams)
  midDelay = average (DParam.midDelays in
listDParams)
  highDelay = average (DParam.highDelays in
listDParams)
  maxBW = average (DParam.maxBW in
listDParams)
END For

```

위와 같이 제안한 알고리즘을 이용함으로써 Bidirectional asymmetric star 구조의 모든 spoke link의 QoS 파라미터가 결정되고 다른 star 구조와 같은 방법으로 복합 구조를 만들 수 있다. 이제 남은 것은 망에서의 QoS 파라미터 변동에 따른 파라미터 값의 갱신 문제인데, 갱신은 스타 구조의 특성상 비교적 적은 계산량을 가지며, 주기적인 시간에 따라 갱신하거나 일정 범위 이상의 변동이 있을 때 갱신하고, 갱신된 토폴로지 요약 정보는 PTSE를 통해 다른 피어 그룹에 전해질 수 있도록 하였다.

3. 시뮬레이션 및 성능 평가

3.1 시뮬레이션 환경

시뮬레이션은 여러 개의 PG와 PG를 구성하는 노드들로 구성되었으며 TA는 단순노드 방식과 복합구성 방식을 적용할 때 그리고 라우팅 시 경로선택에서 링크의 지연시간 또는 대역폭을 기준으로 수행할 경우에 따라 시뮬레이션을 수행하였고 call setup 성공률, 크랭크백(crankback)율, 액세스 시간을 비교하였다. 시뮬레이션을 수행할 네트워크 모델은 3계층의 TA이 가능한 네트워크로 설정하여 최상위 PG가 3

개이고, 각각의 LGN 주소가 "A", "B", "C" 이고 최하위 물리노드 주소는 "A.X.X", "B.X.X", "C.X.X"으로 설정하였다. [표 1]은 시뮬레이션 파라미터를 보여준다. 네트워크 부하는 각 ES에 평균 호 발생을 설정하여 평균 호 발생에 따른 포아송프로세스(poisson process)로 호가 생성된다. 호 설정이 성공하면 호는 지수 분포에 따른 호 지속 시간 동안 호 연결이 수락된다. PTSE 갱신 시간에 따라 각 노드의 토폴로지 데이터베이스의 정확도가 결정된다고 볼 수 있다. PTSE 갱신시간이 짧을수록 데이터베이스 정보의 정확도가 높으나 PTSP 및 플러딩에 따른 노드와 망의 부하가 높아진다. 본 시뮬레이션에서는 PTSE 갱신 시간을 300초에서 900초 사이로 설정하면서 수행하였다.

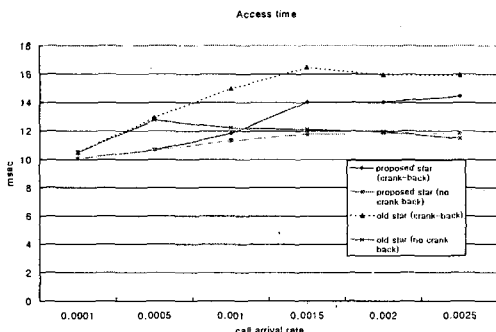
[표 1] 시뮬레이션 파라미터

시뮬레이션 환경변수	값
PTSE refresh time	300~900 sec
Call arrival rate	0.009~0.016
Mean cell period	300 sec
Request bandwidth	1~3 Mbps
Request delay	10~100 Mbps
Max call setup time	100 sec
The number of Max crankback	5회
Link bandwidth	155 Mbps
Link delay	0.0001~0.0007 sec
Simulation time	5000 sec

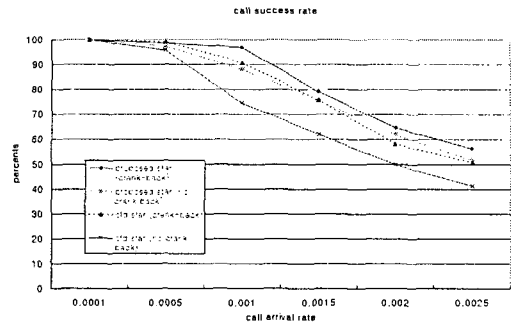
3.2 결과분석

제안한 star TA 알고리즘의 성능을 평가하기 위해, 우선 기존의 star TA 알고리즘의 성능평가를 수행한 후 제안한 TA 알고리즘을 컴퓨터 시뮬레이션을 통하여 성능을 측정하였다. 비교대상의 TA 알고리즘은 기존의 star TA 기법과 본 논문에서 제안한 star TA 알고리즘에 대해 링크의 대역폭, 지연으로 계산하여 구현할 경우에 따라 각각 호 접근 시간, 호 성공률, 크랭크 백 비율의 항목에 대해 비교·평가하였다. [그림 4]와 [그림 5]는 제안된 star TA 기법과 기존 star TA 기법에 대해 크랭크 백이 있는 경우와 없는

경우에 대해 셀 도착율에 따른 접근 시간 및 호 성공율을 나타낸다. 전체적으로 볼 때 크랭크 백이 없는 경우와 크랭크 백이 있는 경우에 비해 많은 차이를 보이고 있는데 이는 크랭크 백이 PNNI 라우팅 프로토콜에서는 아주 중요한 변수가 된다는 것을 알 수 있다. 물론 크랭크 백이 많아지면 네트워크 자원을 많이 소모하고 시그널링 오버헤드가 증가하는 단점이 있지만 그룹 내의 토폴로지 요약정보를 이용하고, 실제의 링크상태 정보를 이용한 경로설정이 가능하지 않다면 크랭크백은 꼭 필요한 과정이라고 할 수 있다. 그림에 보여지듯이 제안된 star 기법이 기존의 기법보다 액세스 시간과 호 성공률에서 더 좋은 성능을 보이고 있는데 이는 제안된 기법이 노드의 라우팅 테이블 정보에서 LGN 내부를 포함한 논리링크의 QoS 파라미터 범위를 가지고 있기 때문에 QoS 제공 가능한 경로를 설정하는데 있어 더 많이 분산시킬 수 있기 때문이다. 특히, 기존 star 구조 TA 기법의 요약정보는 특정링크나 논리링크의 폭주상황에 영향을 많이 받는 단점이 있는데 제안한 다중링크 요약기법을 적용했을 때 그 영향을 많이 줄일 수 있다는 사실을 알 수 있다. 또한 그림에서 각각 크랭크 백이 있는 경우와 없는 경우에 대해서도 비교가 이루어졌는데 크랭크 백이 있는 경우 액세스 시간은 좀더 늘어나게 되고 호의 성공률은 좀더 높아짐을 알 수 있다.



▶▶ 그림 4. 기존 star와 제안된 star의 access time



▶▶ 그림 5. 기존 star와 제안된 star의 호 성공률

4. 결론

본 논문에서 제안한 TA(Topology Aggregation) 기법은 대역폭과 지연 파라미터를 중심으로 피어 그룹내의 토폴로지 정보를 요약하는 기법으로 라인 세그먼트를 이용하여 경계노드 사이의 다중 경로 정보를 요약하였다.

정보량을 줄이고 다중 링크 요약에 유연성을 부여하기 위해 두 개의 라인 세그먼트를 이용하였다. 즉, 하나의 라인 세그먼트를 표현하기 위해서는 양 끝점의 대역폭과 지연 파라미터가 필요하므로 네 개의 파라미터 값이 필요하게 된다. 본 논문에서는 하나의 라인 세그먼트를 표현하기 위해 필요한 네 개의 파라미터 값을 가지고 두 개의 라인세그먼트를 표현하여 다중 링크 요약에 유연성을 부여하였다. 또한, 면적 개념을 도입하여 다중 QoS 파라미터를 하나의 대표 값을 사용하여 여러 가지 TA 기법에 적용하여 계산의 복잡도를 감소시켰다. 그리고, 컴퓨터 시뮬레이션을 통하여 기존의 알고리즘과 비교 분석하였으며 기존의 기법보다 성능이 우수함을 알 수 있었다.

■ 참고 문헌 ■

- [1] Private Network-Network Interface Specification Version 1.0(PNNI 1.0), ATM Forum af-pnni-0055.000, Mar, 1996.
- [2] A.R.Ragozini, et al, "Analysis of the Performance

- of a Hierarchical PNNI Network', Proceedings of the 1999 2nd International Conference on ATM, 1999.
- [3] W. Lee, "Topology Aggregation for Hierarchical Routing in ATM Networks", Computer Communication Review, Apr. 1995.
- [4] Erol Basturk, Paul Stirpe, "A Hybrid Spanning Tree Algorithm for Efficient Topology Distribution in PNNI", Proceedings of the 1998 1st IEEE International Conference on ATM, 1998.
- [5] T. Korkmaz, M. Krunz, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical ATM networks", IEEE, pp.137-146, 1999.