

(19)
(12)

(KR)
(A)

(51) 。 Int. Cl. 7
H04L 12/28

(11)
(43)

2002 - 0068270
2002 08 27

(21) 10 - 2002 - 0007731
(22) 2002 02 09

(30) 09/921,989 2001 02 20 (US)

(71) 10504

(72) 10514 7
11365 169 77 - 35
10591 267
06140 - 2
10977 31
10589 1

(74)

:

(54)

(ASP : Application Service Provider) (WWW) (host) (TCP : Tra
nsfer Control Protocol) (server farm) 가
(provider) 가
(SLA ; service level agreement) 가
(,) TCP

TCP
 TCP (regulator)
 가 , TCP SLA
 () (SLA)
 TCP 가

1

, , , , TCP

1

2 1

(gatekeeper)

3 2

(assitant)

4 2
rate - based algorithm).

(target -

5 4

< >

10 :

12 :

14 :

16 :

18 :

20 :

22 :

24 :

26 :

28 :
 30 :
 34 : 가
 36 :
 38 :

(world - wide networks) ,
 (service provider) (host)
 (Internet World Wide Web site) (global Internet)

가 , 가
 () (outsourcing)
 (ASP ; Application Service Provider) ASP
 (server farm))
 (" " (hosted customers) "
 ") 가 . ASP

(, (IP)) , 가
 (rotocol) (TCP; Transmission Control P
 SLA TCP , i j TCP Nmin(i,j)
 TCP Nmax(i,j) . TCP Nmin(i,j) TC
 P , TCP Nmin(i, Nmin(i,j)
 j) 가 가 (sharable capacity) 가 , 가 ,
 가 i j (operator) j 가
 , j TCP
 가 TCP , Nmin(i,j) Nmax(i,j) (best
 - effort basis) TCP , TCP Nmin(i,j)

Nmin(i,j) TCP Nmin(i,j) TCP

P) (Lucent/Xedia : www.xedia.com) (Access Point : A)
 (PhaseCom : www.speed - demon.com) (Speed - Class)
 (inbound traffic ; SLA (bandwidth usage - based SLA)
 가 TCP 09/543,207 [; YO999 - 374]
 (Nmin(i,j),Nmax(i,j)) TCP SLA

SLA , TCP
 SLA

(TCP)가 (ASP)
 SLA 가 (,) TCP
 (operation flexibility) 가

1
 TCP IP (, TCP TCP SLA)
 (rate admittance) ()
 (SLA) 가 SLA (,)
) TCP TCP TCP TCP
 가 TCP TCP

(common buffer) IP (FIFO ; first in first out)

SLA () TCP TCP
TCP SLA TCP
가

() TCP SLA ASP ASP
가

가 가 TCP (being stateless)" (TCP
TCP " " " " TCP TCP
TCP (initiator) TCP TCP

1 WWR (12) (20)
TCP TCP (10) 1 () (14) WWR(12) IP (W
WR(12) IP (TCP TCP)
(admitted traffic) () (HS - LAN; 16) (18)
TCP WWR 가 (34) , WWR 가 (34)
(outbound link) (22) (16)
(18) " IBM(International Business Machines : www.ibm.com)" 가
(SecureWay Network Dispatcher) , (Cisco Systems : www.cisco.com)
가 (LocalDirector) (workload dispatching product)
(18) (20) (S1, S2, ... Sn) (spray
ing)

WWR(12) (28), (30), 가 (34), (36), (38) 가
(28) TCP () (14) (28) , (30)가
(30)가 SLA TCP
() (30)가
(16) (18) (30)가
가 (34) , 가 (34) TCP
가
(22) TCP RST(reset) ON TCP ()

(38) 가 TCP (30)가 가 TCP (30) (30)가 가 TCP (38) (30)가 , (38)가 . (30)가 가 TCP (38) (30) , WWR(12) (,) TCP SLA (24) , (36) SLA (24) (26) (40) WWR(12) , (26) (40) . (30), 가 (34) (38)

$Nlimit(j)$ (20) j TCP

$Nmin(i,j)$ i j TCP

$Nmax(i,j)$ $Nmin(i,j)$

$Nadmitted(i,j)$ i j (20) TCP

$Ntotal(j)$ i $Nadmitted(i,j)$

$Nmin(j)$ i $Nmin(i,j)$

$Nexcess(j)$,

$Nexcess(j) = Nlimit(j) - Nmin(j)$

$Nunused(j)$; i $Nmin(i,j) > Nadmitted(i,j)$ 가 $[Nmin(i,j) - Nadmitted(i,j)]$

$Nsharable(j)$ $Nmin(i,j)$ TCP ; $Nsharable(j) = Nexcess(j) + Nunused(j)$.

$Ntarget(i,j)$ (i,j) TCP

: (30)가 TCP , " " 가 " "

2 (30) . 1 4 (28)

, C 가 , (38)(3 1) Nunused , Cadmitted Nadmitted 가 , Crejected Nrejected 가

, , Nadmitted Nrejected 0 , Nsharable Nexcess Nunused
 2 (28)
 TCP SYN 가 ON
 SYN 가 OFF (TCP TCP SYN 가
 ON (TCP), 3 .

3 Nmin 3
 (i,j)가 TCP (Nadmitted(i,j))가 (Nmin(i,j)) TCP
 sharable) 4 , (Nmin(i,j)) TCP 가 (N
 ted(i,j)) Nsharable(j) 0 TCP 가 (34)

3 2 (38)
 unused = 0 가 " " N
 " , 3 가 "
 , 2 1 Nunused(j)
 TCP " , WWR(12)
 Nunused(j) Unused_Permit_Factor(j)
 (constant multiplier) 0 1

가 (34) (, 4) 2 TCP
 (RST) ON TCP , 가 (34) TCP
 (immediate closure) (2). 2 TCP (1), TCP
 , 2 " (amicable)"

WWR(12) TCP
 , 가 , 4 2 (30)
 (38) Ntarget(i,j) 4 1
 () . 2 Ntarget(j) SYN 가 OFF
 mitted(j)가 4 3 TCP , TCP 가 Nad
 34) (

5 (38) 4

(i,j) Ntarget(i,j) 5
 TCP 4
 (STOP) Ntarget(i,j)가 1 Ntarget(i,j)가
 j) Nmax(i,j) 2 (i,j) Ntarget(i,
 t(i,j)가 Nmax(i,j) 3 가 " " Ntarget(i,
 Nsharable(j) Ntarget(i,j)가 Nmin(i,j) Ntarget
 (i,j) Nsharable(j) Ntarget(i,j)가 Nmin(i,j) (i,j) , Ntarget(
 i,j) Nmax(i,j) (, 4

, WWR(12) , (,) TCP
 SLA
 (40) 가 WWR(12) (20)
 가 ,
 , ASP
 가 WWR(12) ,
 Nexcess(j) , Unused_Permit_Factor(j) ,
 TCP 가
 가 " (wire)" (, WWR(12) " " ,
) , WWR(12) TCP

WWR(12) , , WWR(12)
 , 가
 , 가
 , 가
 가 TCP SLA ,
 가 ,
 ,
 가 ,
 가 TCP

(57)

1.

TCP
(service level agreement) -

TCP
(host) - ,

TCP TCP IP - ,

TCP , TCP

TCP .

2.

1 ,

IP TCP TCP ,

3.

2 ,

TCP TCP , TCP IP

TCP IP TCP TCP ,

4.

가 - , TCP TCP

TCP TCP IP - ,

TCP ,

TCP , IP TCP

IP

IP TCP TCP ,

TCP IP ,

TCP , TCP , TCP

TCP

5.

4 ,

(revision) ,

6.

TCP TCP

TCP

CP TCP IP , T

TCP TCP

7.

6 ,

TCP TCP

8.

6 ,

(outbound traffic)

TCP

9.

6 ,

IP IP TCP TCP TCP , TCP

10.

9 ,

IP SYN 가 ON OFF TCP

11.

9 ,

TCP TCP TCP , TCP

12.

11 ,

TCP TCP TCP ,

13.

12 ,

CP TCP T

14.

12 ,

TCP ON TCP

15.

6 ,

TCP TCP TCP ,

16.

6

TCP

17.

6

TCP

18.

6

/ (real - time admittance/rejection algorithm)

TCP

TCP

19.

6

/ (target - rate - based admittance/rejection algorithm)

TCP

TCP

20.

가 - , - TCP TCP

TCP

TCP

IP

- ,

-

TCP

TCP

IP , IP TCP IP TCP , TCP TCP , TCP TCP TCP , TCP TCP

21.

20 ,

22.

20 ,

IP SYN 가 ON OFF

23.

20 ,

TCP 가

24.

20 ,

ON TCP 가

25.

20 ,

TCP TCP ,

26.

20 ,

(manager means)

27.

20 ,

/ ,

TCP

.

28.

20

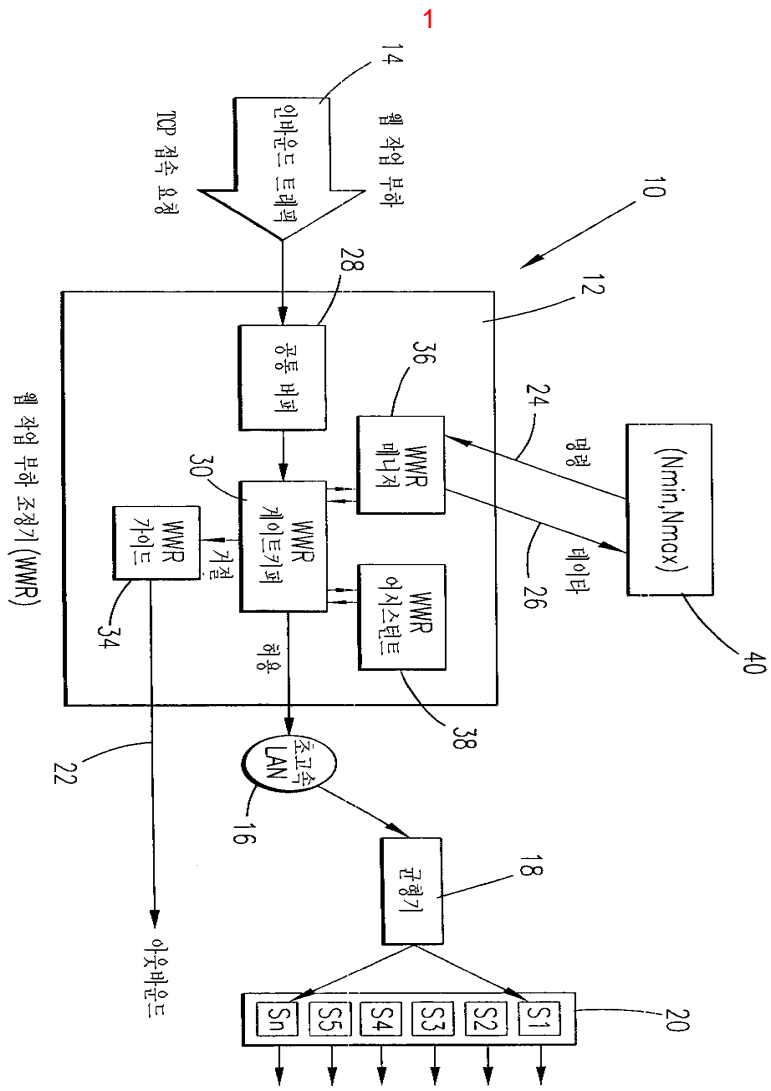
,

/

,

TCP

.



```

REPEAT:
(단계 1) --If the cycle-time has expired, /*변수들을 재설정함으로써 다음 싸이클에 대하여 준비한다*/
then do; C = C + 1;
        Get Nnumused (j) from WWR Assistant;
        set Cadmitted(i,j) = Cadmitted(i,j) + Nadmitted(i,j) for all (i,j);
        set Crejected(i,j) = Crejected(i,j) + Nrejected(i,j) for all (i,j);
        set Nadmitted(i,j) = 0 for all (i,j); set Nrejected(i,j) = 0 for all (i,j);
        set Nsharable(j) = Nexcess(j) + Nnumused(j) for all (j);
        end;
(단계 2) --Get a packet;
        If the packet's SYN bit is OFF
        then do: let the packet pass through; go to REPEAT; end;
        /* TCP 접속 요청을 더룬다 */
        Find its index (i,j);
(단계 3) --If Nadmitted(i,j) < Nmin(i,j) /*최소 SLA를 보장한다 */
        then do: set Nadmitted(i,j) = Nadmitted(i,j) + 1;
        let the packet pass through;
        go to REPEAT;
        end;
(단계 4) --If (Nadmitted(i,j) >= Nmin(i,j) & (Nadmitted(i,j) < Nmax(i,j) & (Nsharable(j) > 0)
        /*요청이 허용될 수 있다 */
        then do; set Nadmitted(i,j) = Nadmitted(i,j) + 1; Set Nsharable(j) = Nsharable(j) - 1;
        let the packet pass through;
        go to REPEAT;
        end;
        /* 요청이 거절되어야 한다 */
        else do; set Nrejected(i,j) = Nrejected(i,j) + 1;
        forward the packet to WWR Guide;
        go to REPEAT;
        end;

```

2

3

```

set Numused(j) = 0 for all (j);
If the mode of operation is "not borrowing" then return;
/*여기서는 동작 모드가 "차용"으로 되고, 따라서 모든 j에 대하여 Numused(j)를 계산할 필요가 있다*/
For every (j) and for every (i),
  If Nadmitted(i,j) < Nmin(i,j) then set Numused(j) = Numused(j) + (Nmin(i,j) - Nadmitted(i,j));
For every (i), Numused(i) = Numused(i) * Unused_Permit_Factor(i);
return;

```

```

/* 비율에 기초한 게이트키퍼 알고리즘 */
REPEAT:
---- If the cycle-time has expired, /*변수를 재설정함으로써
    다음 사이클에 대해 준비한다*/
    then do; C = C + 1;
        set Cadmitted(i,j) = Cadmitted(i,j) + Nadmitted(i,j) for all (i,j);
        set Crejected(i,j) = Crejected(i,j) + Nrejected(i,j) for all (i,j);
        set Nadmitted(i,j) = 0 for all (i,j); set Nrejected(i,j) = 0 for all (i,j);
    end;

(단계 1)

----Get a packet;
    If the packet's SYN bit is OFF
        then do: let the packet pass through; go to REPEAT; end;
        /* TCP 접속 요청을 다룬다 */
        ----Find its index (i,j);
        IfNadmitted(i,j) < Ntarget(i,j)
            /* 요청을 받아들인다 */
            then do; set Nadmitted(i,j) = Nadmitted(i,j) + 1;
                let the packet pass through;
                go to REPEAT;
            end;
            /* 요청을 거절한다 */
        else do; set Nrejected(i,j) = Nrejected(i,j) + 1;
            forward the packet to WWR Guide;
            go to REPEAT;
        end;

(단계 2)

4
(단계 3)

```

5

```

/*Ntarget(i,j)를 계산한다*/
(단계 1) --- Do the following for every application class j:
                /*어떠한 요청 기원도 일어나지 않았다*/
                If (Nrejected(i,j) = 0) /*어떠한 SJA 위반도 감지되지 않았다. 여기서 <=는 작거나 같음을 의미한다*/
                    & (Nadmitted(i,j) <= Nmax(i,j)) over all i
                    then go to STOP;
                COMPUTE_TARGETS:
(단계 2) --- Let Ntarget(i,j) = Nadmitted(i,j) + Nrejected(i,j) for all i;
                Compute Ntotal(i); /*여기서 Ntotal(i)는 모든 i에 대하여 Ntotal(i, j)의 합이다*/
                For every (i,j) such that Ntarget(i,j) > Nmax(i,j) /*수요가 최대치를 초과한다*/
                    first set Ntotal(i) = Ntotal(i) - (Ntarget(i,j) - Nmax(i,j)); /*중요를 조절한다*/
                    And then set Ntarget(i,j) = Nmax(i,j); /*수요가 최대치를 초과하지 못하게 한다*/
                    If Ntotal(i) <= Nlimit(i,j) then go to STOP;
                /*각 Ntarget(i, j)는 Nmax(i, j) 미만으로 설정될 필요가 있다*/
(단계 3) --- Let Nexceed(i) be the sum of (Ntarget(i,j) - Nmin(i,j)) over those i such that Ntarget(i,j) > Nmin(i,j);
                If the node-of-operation is "non borrowing" then set Nsharable(i) = Nexcess(i);
                If the node-of-operation is "borrowing" then compute Numused(i) and set Nsharable(i) = Nexcess(i) +
                Numused(i) * Unused_Permit_Factor(i);
                /*다음에 수행되는 것은 공평한 할당이다. 그러나, 우선 순위에 기초한 할당이 대신 수행될 수도 있다*/
                For every i such that Ntarget(i,j) > Nmin(i,j)
                    Set Ntarget(i,j) = Nmin(i,j) + (Ntarget(i,j) - Nmin(i,j)) * (Nsharable(i) / Nexceed(i));
                    else set Ntarget(i,j) = Nmax(i,j);
(단계 4) --- STOP;
                End;

```