

# Differentiated Services에서 TCP의 ECN을 이용한 성능향상에 관한 연구

오 종 채    정 재 일  
한양대학교 전자전기공학부

## Study on TCP ECN's Capability of Improving the Performance of Differentiated Services Architecture

Jong Chae Oh    Jae il Jung  
Dept. of Electrical and Computer Engineering, Hanyang University

### Abstract

Today, QoS is one of the most critical issues in the network research area and Differentiated Services (DiffServ) is considered as the most prominent solution to provide some kinds of service differentiation without introducing any scalability problem. Among DiffServ's approaches, Assured Service (AS) provides some minimal level of QoS guarantee by treating more preferably than traditional Best Effort (BE) traffic and by using different level of drop probabilities within the same AS classes. In this paper, we investigate the ECN's capability of improving overall goodput of the flows and the possibility of resolving the fairness problem among the flows belonging to same class in Differentiated Services architecture.

### I. 서론

그동안 인터넷상에서 QoS를 제공하기 위한 많은 연구가 진행되어 왔다. 최근에 DiffServ는 현재의 인터넷구조에서 확장하기 쉽고 구조가 비교적 간단하기 때문에 가장 유망한 모델로 여러 분야에서 연구가 진행되고 있는데, Assured Service분야에서는 같은 클래스에 속하는 flow들 간에 대역폭을 공평하게 할당하는 문제가 대두되었다[1]. 본 논문에서는 TCP에서 향상된 흐름제어(Flow Control)을 위해 제안된 ECN(Explicit Congestion Notification)을 이용하여 이 문제에 접근함과 동시에 goodput관점에서의 성능향상을 가져올 수 있음을 확인하고자 한다.

본 논문의 구성은 다음과 같다. 2절에서는 ECN(Explicit Congestion Notification)과 DiffServ의 Assured Service에 대해서 간략히 기술한다. 3절에서는 DiffServ에 ECN을 RED와 RIO알고리즘에 어떻게 적용했는지에 대해서 설명한다. 4절에서는 시뮬레이션 환경과 결과에 대해 분석하고 마지막으로 5절에서는 향후 연구방향에 대해서 언급하도록 하겠다.

### II. ECN, RED, and RIO

#### 2.1 Explicit Congestion Notification(ECN)

Explicit Congestion Notification(ECN)[2]은 네트워크상에 일어날 가능성이 있는 혼잡을 미리 탐지하여 이를 송신측에 알려 전송속도를 조절하는 방식으로, RED 알고리즘이 임박한 혼잡을 탐지했을 때 임의의 패킷들을 폐기하는 것과 달리, ECN알고리즘은 혼잡이 예상되면 도착하는 임의의 패킷의 IP헤더에 존재하는 CE-bit(Congestion Experienced bit)을 세팅(setting)하고, 이 패킷이 수신측에 도착하면 이를 송신측에 알리기 위해서 생성하는 ACK 패킷의 TCP헤더에 존재하는 ECN-Echo 플래그를 세팅하여 전송한다. 이 플래그의 세팅은 이후 도착하는 같은 flow의 패킷 중에서 CWR(Congestion Window Reduced)플래그가 세팅되어 도착할 때까지 계속된다. 송신측에서는 ACK패킷의 ECN-Echo 플래그가 세팅되어 도착하면 혼잡이 일어날 수 있음을 감지하고 송신윈도우의 크기를 절반으로 줄이는 등의 TCP Reno 메커니즘과 유사한 대응책을 취한다. 그리고, 다음 송신 패킷의 CWR 플래그를 세팅하여 보냄으로써 적절한 대책을 취했음을 수신측에 알려 더이상 ECN-Echo 플래그를 세팅하지 않도록 한다. IP 레벨에서 IP 헤더에 존재하는 두 개의 bit가 혼잡상황을 알리는데 이용된다. 하나는 CE(Congestion Experienced)-bit이고, 다른 하나는 ECT(ECN Capable Transport) bit이다. 이 논문의 동기를 제공한 것은 이 두 개의 bit가 바로 DiffServ의 DS field에 존재하는 마지막 두 bit인 CU(Currently Unused) field라는 사실이다[4,5]. 이는 ECN 기능을 DiffServ구조에 적용시키는데 별다른 충돌과 오버헤드 없이 투명하고 손쉬운 호환성을 제공할 수 있음을 의미한다. TCP 레벨에서는 위에서 설명한 ECN-Echo 플래그와 CWR 플래그가 ECN을 구현하기 위해서 사용되는데, 이 두 개의 플래그는 기존 TCP 헤더의 option 필드에 쉽게 정의할 수 있어 손쉽게 구현될 수 있다.

## 2.2 Best Effort Service and Assured Service in Differentiated Services Architecture

DiffServ내에서 Best Effort Service는 default PHB(000000)값을 가지고 오늘날의 일반 인터넷 트래픽과 같은 방식으로 서비스되고, 이때 혼잡 제어 방식으로 주로 RED 알고리즘이 사용되고 있다. 한편 EF PHB(6)를 이용하여 최소한 정해진 수준 이상의 QoS를 보장함으로써 실시간 서비스 등에 적합한 전송품질을 보장하는 Premium Service와는 달리 Assured Service는 명확히 규정된 QoS를 보장하지는 못하나, 서비스 프로필에 근접한 수준의 서비스 품질을 제공한다. Assured Service는 AF PHBs[7]를 이용하여 구현되는데, AF PHB는 3개의 drop precedence를 가진 4개의 서로 다른 서비스 클래스로 구성된다. Assured Service의 핵심 구성요소는 RIO(RED with In and Out) queue 알고리즘이다. RIO는 RED를 확장한 것으로 In-profile 패킷과 Out-of-profile 패킷을 구별하여 서비스 한다. DS-domain의 Ingress 라우터에서는 각 트래픽 flow의 프로필에 의거한 marking이 행해지는데, 이 프로필의 준수여부에 따라 In-profile과 Out-of-profile로 구별되어 다른 drop precedence를 갖는 PHB를 가지게 된다. 각 flow의 프로필은 SLA(Service Level Agreement)를 통하여 정의된다. Marker로는 토큰 버킷 알고리즘이 간단하면서도 Assured Service에 적당한 것으로 판별되었다[1]. RIO는 3개의 drop precedence를 크게 두 가지(IN/OUT)로 분류하여 적용하는 반면에 3개의 drop precedence를 구별하여 적용하는 Three Color Marker 방식도 존재한다. 본 연구에서는 RIO를 이용하여 Assured Service를 구현한다.

### III. Modification of RIO and RED

BE 트래픽에 적용되는 RED 알고리즘과 AF 트래픽에 적용되는 RIO 알고리즘에서 Early dropping은 average queue length가 미리 정의된 minimum threshold를 넘어서면 발생하게 된다. 그리고, average queue length가 증가할수록 버려질 패킷을 선택하는 drop probability 역시 증가하여 더 많은 수의 패킷이 폐기되고, maximum threshold를 넘어서게 되면 도착하는 모든 패킷들이 queue size가 maximum threshold 이하로 감소할 때까지 폐기된다. RIO는 In-profile packet과 Out-of-profile packet에 서로 다른 별도의 RED 알고리즘을 적용하는데, threshold parameter와 drop probability, 그리고 이 확률의 증가비율이 다르게 적용된다. 추가적으로 average queue length를 계산하는 알고리즘에도 차이가 있는데, Out-of-profile packet에 적용되는 알고리즘은 queue에 도착한 모든 패킷의 수를 고려하는 반면에 In-profile 패킷에 적용하는 알고리즘에는 도착한 In-profile 패킷의 수만 계산에 이용된다.

본 논문에서는 average queue length가 minimum threshold를 넘어서게 되면 도착하는 패킷의 CE-bit 을 세팅하고, 아울러 그 패킷의 drop precedence 값을 낮춘 다음 폐기하는 대신 queue에 저장한다. 따라서 average queue size가 두 threshold값 사이에 있을 때는 dropping이 일어나지 않는다. Average queue size가 maximum threshold를 넘어서는 심한 혼잡의 경우에는 기본 알고리즘과 마찬가지로 도착하는 모든 패킷들을 queue size가 감소할 때까지 폐기하게 된다.

### IV. 실험 결과 및 검토

시뮬레이션은 ns-2[8] 시뮬레이터와 Sean Murphy의 DiffServ-patch[9]를 이용해서 행해졌다. ns-2에는 기본

ECN 기능이 구현되어 있으나, DiffServ-patch에는 이 기능이 전혀 수용되어 있지 않아 필요한 파일들을 수정하여 DiffServ에 ECN 기능을 부여하였다.

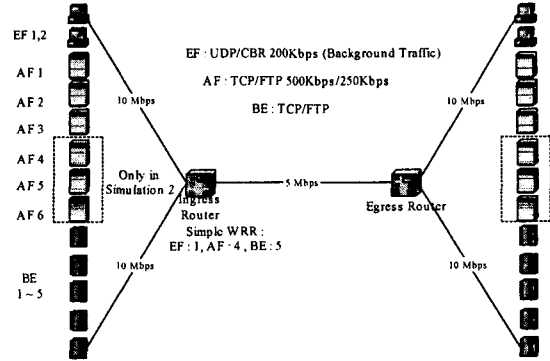


그림 1. 시뮬레이션 모델

Simulation 1에는 2개의 EF 트래픽이 background 트래픽으로서 존재하고, 같은 클래스에 속하는 3개의 AF 트래픽과 5개의 BE 트래픽이 존재한다. EF 트래픽은 200Kbps의 속도로 전송하는 UDP/CBR 소스를 사용하고 AF와 BE 트래픽 소스는 무한 FTP application으로 기본 TCP/Reno의 특성을 지닌다. EF 트래픽의 수신측은 단순히 패킷을 받아들이는 Null agent이고, 나머지 AF와 BE의 수신측은 TCP sink agent로서 도착하는 모든 패킷들에 대해 ACK packet을 생성하는 기본 TCP/Reno agent이다. TCP agent 들에는 ns-2의 기본 파라미터 값들을 사용했으며, 비교를 목적으로 DiffServ-patch에 포함된 예제와 같은 topology와 parameter를 가지고 시뮬레이션을 행하였다. Scheduler는 WRR(Weighted Round Robin) 알고리즘을 사용하는데, 3개의 트래픽 클래스(EF, AF, BE)에 각각 1, 4, 5의 weight값을 적용한다.

Simulation 1과 2의 차이점은 AF 트래픽의 수신데, simulation 2에서는 AF 트래픽들간의 관계를 좀 더 자세히 살펴보기 위해서 6개의 AF 트래픽을 사용하였다. 따라서 각 AF 트래픽의 target rate는 절반으로 적용하였다. 논문에서 사용된 시뮬레이션 환경을 나타내기 위해 다음과 같은 용어를 사용한다.

표 1. 세 가지 simulation 시나리오 표기 용어

no ecn	ECN 기능이 적용되지 않은 시나리오
ecn to AF	AF 트래픽에만 ECN이 적용된 시나리오
ecn to AF, BE	AF와 BE 모두에 ECN이 적용된 시나리오

표 2. Queue parameters of RED and RIO

	queue weight	min_th	max_th	*max P <sub>drop</sub>	min_th	max_th	max P <sub>drop</sub>
RED	0.002	50	180	0.05			
RIO	0.002	for OUT packets			for IN packets		
		20	40	0.04	45	90	0.02

\*max P<sub>drop</sub> : maximum drop probability, Simulation Time:100sec  
 RED queue size : 200, RIO queue size : 100 (unit : packets)

#### 4.1 Simulation 1

2개의 EF 트래픽, 500Kbps의 peak-rate profile을 가진 AF 트래픽 3개, 그리고 5개의 BE 트래픽이 존재한다. 각 노드로부터 라우터까지의 link delay는 0.1ms이고(Source node-Ingress Router, Egress Router-Destination node),

두 라우터간의 link delay는 1ms이다.

4.1.1 Goodput and Fairness among the AF flows in the same class

그림 2에는 'no ecn' 과 'ecn to AF' 시나리오의 시뮬레이션을 통한 goodput결과가 비교되어 있다. ECN 기능을 이용했을 때, 향상된 결과를 볼 수 있다. 이와 함께 확인할 수 있는 중요한 사실은 3개 flow 사이의 goodput 편차인데, 그림 2에 나타나 있듯이 ECN을 사용했을 때 goodput의 표준편차가 상당히 감소했음을 알 수 있다. EF에는 500Kbps가 할당되나 2개의 flow는 각각 200Kbps로 제한되기 때문에 100Kbps의 잉여 대역폭은 AF와 BE에게 할당되게 된다. 그런데, 'no ecn'의 경우 가장 적은 goodput을 얻는 2번 flow는 이 여분의 대역폭이 존재함에도 AF에 할당된 기본 대역폭인 2000Kbps의 1/3인 667Kbps보다도 적은 660.7Kbps를 얻는데 그쳤다. 그러나 ECN을 적용한 경우에는 최소대역폭이 669.6Kbps로 증가하고 최대대역폭도 691.7Kbps로 감소하여 같은 AF 클래스에 속한 flow들간의 편차가 확연히 감소함을 알 수 있다. 이는 ECN에 의해 압박한 혼잡의 발생 가능성을 초기에 알림으로써 극심한 혼잡을 예방하여 전체적인 goodput을 증가시키고 동시에 부수적으로 flow들간의 fairness를 향상시켜줄 수 있음을 보여준다.

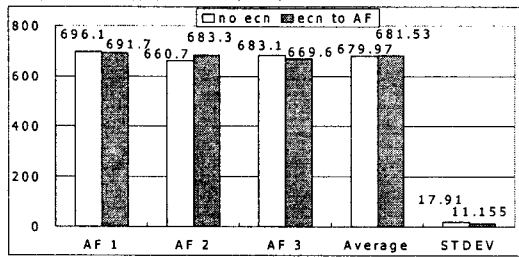


그림 2. Average and Standard Deviation of Goodput of AF flows

4.1.2 ECN's Effects on the relationship between AF and BE flows

위에서 살펴본 두개의 시나리오 이외에 BE 트래픽에도 ECN을 적용시키는 시나리오('ecn to AF, BE')를 추가하여 AF 트래픽과 BE 트래픽 사이에 발생하는 현상을 살펴 보았다. 결과는 매우 흥미로운데, BE 트래픽에 ECN을 적용하면 BE flow들 사이에서도 fairness가 향상됨을 알 수 있는데 이 때문에 AF flow들간의 fairness가 손상되는 결과를 낳는다. 이 시나리오에서 BE 트래픽의 경우에는 fairness와 goodput 모두 상당히 향상된 결과를 나타내는데 이는 ECN 기능이 클래스에 관계 없이 같은 클래스에 속하는 flow들간의 fairness를 향상시키는 일관된 능력이 있음을 보여주고 있다.

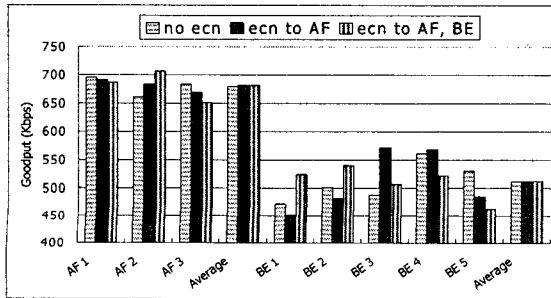


그림 3. Average Goodput of AF and BE flows

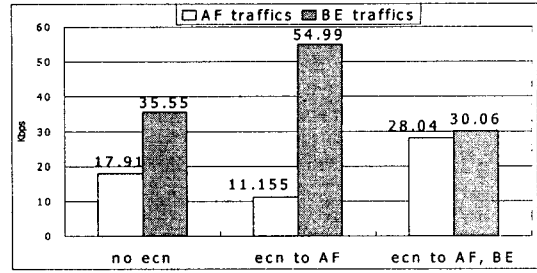


그림 4. Standard Deviation of Goodput of AF and BE flows

4.1.3 OUT/IN packet ratio, Early drop ratio, and CE-bit marking ratio

표 3은 Ingress router의 buffer에서 측정된 몇 가지 통계치를 보여 준다. ECN 기능을 추가하지 않았을 때, Out-of-profile packet의 비율이 상당히 높는데 이는 송신측이 혼잡 상황이 일어나고 있음을 3개의 중복된 ACK 패킷을 수신하거나, 재전송타이머(Retransmission Timer)를 넘어서도록 ACK 패킷이 도착하지 않았을 때라야 비로소 알아차리기 때문에 실제로 혼잡이 발생하는 시간과 송신측에서 인지되는 시간사이에 계속해서 전송된 패킷들은 결국 OUT 패킷으로 marking되기 때문이다. 표 3에서 'forced drop' 이란 queue size가 maximum threshold를 넘어서서 폐기되는 현상을 지칭한다. 전체적으로 IN 패킷은 drop이나 CE-bit marking이 발생하지 않으며, 'no ecn' 시나리오에서 forced drop은 발생하지 않으나 early drop의 비율이 거의 3%에 이르는 반면 ECN을 적용한 경우 marking 비율은 early drop 비율보다 높은 반면 전체적인 drop비율은 상대적으로 매우 낮음을 알 수 있다.

표 3. Ingress Router에서의 통계값

	no ecn	ecn to AF	ecn to AF, BE
Average OUT/IN packets ratio(%)	31.93	27.88	27.75
OUT packets early drop ratio (%)	2.92	.	.
OUT packets forced drop ratio (%)	.	0.5	0.6
OUT packets CE-bit Marking ratio (%)	.	4.38	4.92

4.2 Simulation 2

simulation 1과 비교하여, AF 트래픽이 6개로 늘어나고, 각 프로파일의 peak rate은 절반인 250Kbps로 감소하였고 나머지 환경 설정은 변화 없다. 따라서, 전체 bottleneck 대역폭의 각 트래픽 클래스에의 할당비율 역시 변화 없다.

4.2.1 Goodput and Fairness among the AF flows in the same class

Simulation 1과 비교하여 볼 때, 전체적인 goodput은 상대적으로 많이 향상되었으나, 나타난 결과치로는 fairness 면에서는 ECN을 사용하지 않았을 때에 비해 별다른 성능 향상을 보여주지 않는다. 그러나, 좀 더 자세히 살펴보면 'no ecn'의 경우에는 333.3(2000/6)Kbps이하의 대역폭을 얻는 flow가 4개나 존재하는 반면에 'ecn to AF'의 경우에는 flow 4를 제외한 나머지 모든 flow들이 그 이상의 대역폭을 할당받는다. 단, AF flow 1이 상대적으로 많은 대역폭을 할당받은 결과 수치상의 표준편차가 약간 증가했을 뿐 전체적으로 AF flow들은 ECN을 사용한 결과 끌고루 기본 대역폭 이상을 사용함을 알 수 있다.

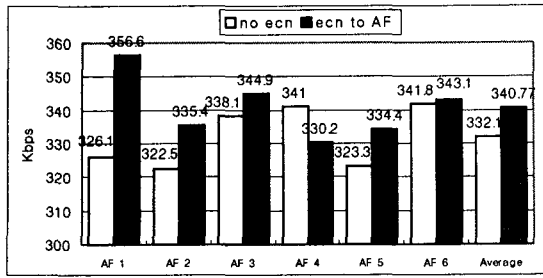


그림 5. Average Goodput of the AF flows

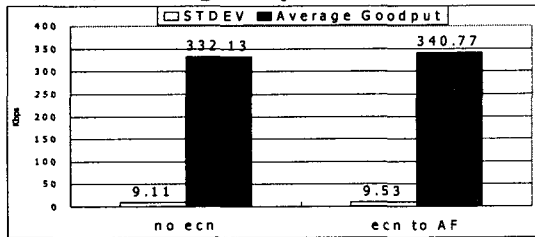


그림 6. Standard Deviation and Average Goodput of AF flows

#### 4.2.2 ECN's Effects on the relationship between AF and BE flows

그림 8과 9는 simulation 1에서의 결과와 비슷한 결과를 나타내고 있다. ECN을 BE 트래픽에 적용함으로써 향상된 fairness와 goodput을 얻을 수 있다. 그러나 simulation 1에 비해서 BE 트래픽에 ECN을 적용함으로써 발생하는 AF 트래픽에의 악영향은 상당히 감소했음을 볼 수 있다. 이는 AF flow의 수가 증가했기 때문에 각 flow에의 영향이 감소했기 때문으로 볼 수 있다.

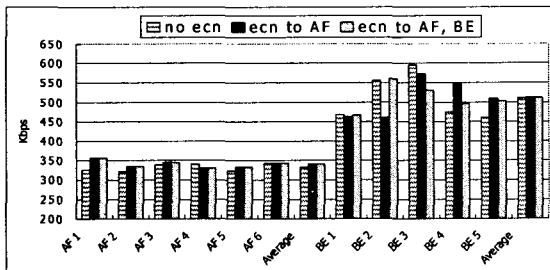


그림 8. Average Goodput of AF and BE flows

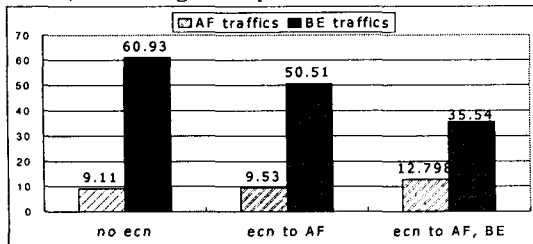


그림 9. Standard Deviation of Goodput of AF and BE flows

#### 4.2.3 OUT/IN packet Ratio, Early drop ratio, Forced drop ratio and CE-bit Marking ratio

표 4는 표 3과 전체적으로 유사한 결과를 나타낸다. 그러나, simulation 1과는 달리 모든 시나리오에서 IN 패킷

(in-profile packet)의 경우에 forced drop이 존재한다는 점이 차이를 보인다. 표에 나타나 있듯이 전체적인 OUT/IN 비율의 증가는 flow수의 증가로 인한 것으로 각 송신측이 프로필에 정의된 것보다 많이 전송하려 시도했음을 나타내고, 따라서 Ingress 라우터의 버퍼가 maximum threshold를 넘어서는 현상이 빈번하게 발생했기 때문에 나타난 현상으로 보인다.

표 4. Ingress Router에서의 통계값

	no ecn	ecn to AF	ecn to AF, BE
Average OUT/IN packets ratio(%)	39.02	32.86	32.8
IN packets forced drop ratio(%)	0.043	0.041	0.041
OUT packets early drop ratio(%)	10.9	.	.
OUT packets forced drop ratio(%)	0.14	0.2	0.2
OUT packets CE-bit Marking ratio(%)	.	20.17	18.52

### V. 결론

본 논문에서 우리는 TCP의 ECN 기능이 DiffServ구조에서 같은 트래픽 클래스에 속한 flow들간의 대역폭 할당량을 증가시켜 줌과 동시에 같은 클래스내의 flow들에 대해서 대역폭 할당에 대한 fairness를 향상시켜줌을 확인하였다. ECN을 적용하는데 있어서의 이점은 DiffServ의 발전을 위해 여러가지 측면에서 연구되고 있는 많은 방법들과 별다른 충돌 없이 구현될 수 있다는 점이다. DiffServ 구조에서 ECN의 역할을 좀 더 면밀히 검토하기 위해서는 SACK(Selective ACK)과 같은 다양한 TCP 구현에서의 시뮬레이션 결과를 검토하고, [4]에서 언급된 바와 같이, end-to-end 기반으로 ECN을 사용할 경우 congestion notification에 반응하는 TCP의 mechanism 또한 DiffServ의 고유한 특성을 고려하여 수정되어야 할 것이다.

### Reference

- [1] J. Ibanez and K. Nichols, Preliminary Simulation Evaluation of an Assured Service, draft-ibanez-diffserv-assured-eval-00.txt, Internet Draft, Aug 1998
- [2] S. Floyd, TCP and Explicit Congestion Notification, ACM Computer Communication Review, Vol. 24, No. 5, pp.10-23, Oct 1994
- [3] S.Floyd, and V. Jacobson, Random Early Detection Gateways for Congestion Avoidance IEEE/ACM Transactions on Networking, Vol. 1, No. 4, pp.397-413, Aug 1993
- [4] K. Nichols, S. Blake, F. Baker and D. Black, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, Dec 1998
- [5] K. Ramakrishnan, S. Floyd, A Proposal to Add Explicit Congestion Notification(ECN) to IP, RFC 2481, Experimental RFC, Jan 1999
- [6] V. Jacobson, K. Nichols and K. Poduri, An Expedited Forwarding PHB. RFC 2598
- [7] J. Heinanen et al, Assured Forwarding PHB Group, RFC 2597
- [8] UCB/LBNL/VINT Network Simulator - ns (version 2), <http://www.isi.edu/nsnam/ns/>
- [9] S. Murphy. Diffserv additions to ns-2, <http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/>