Dynamic Admission Control For A Bandwidth Broker

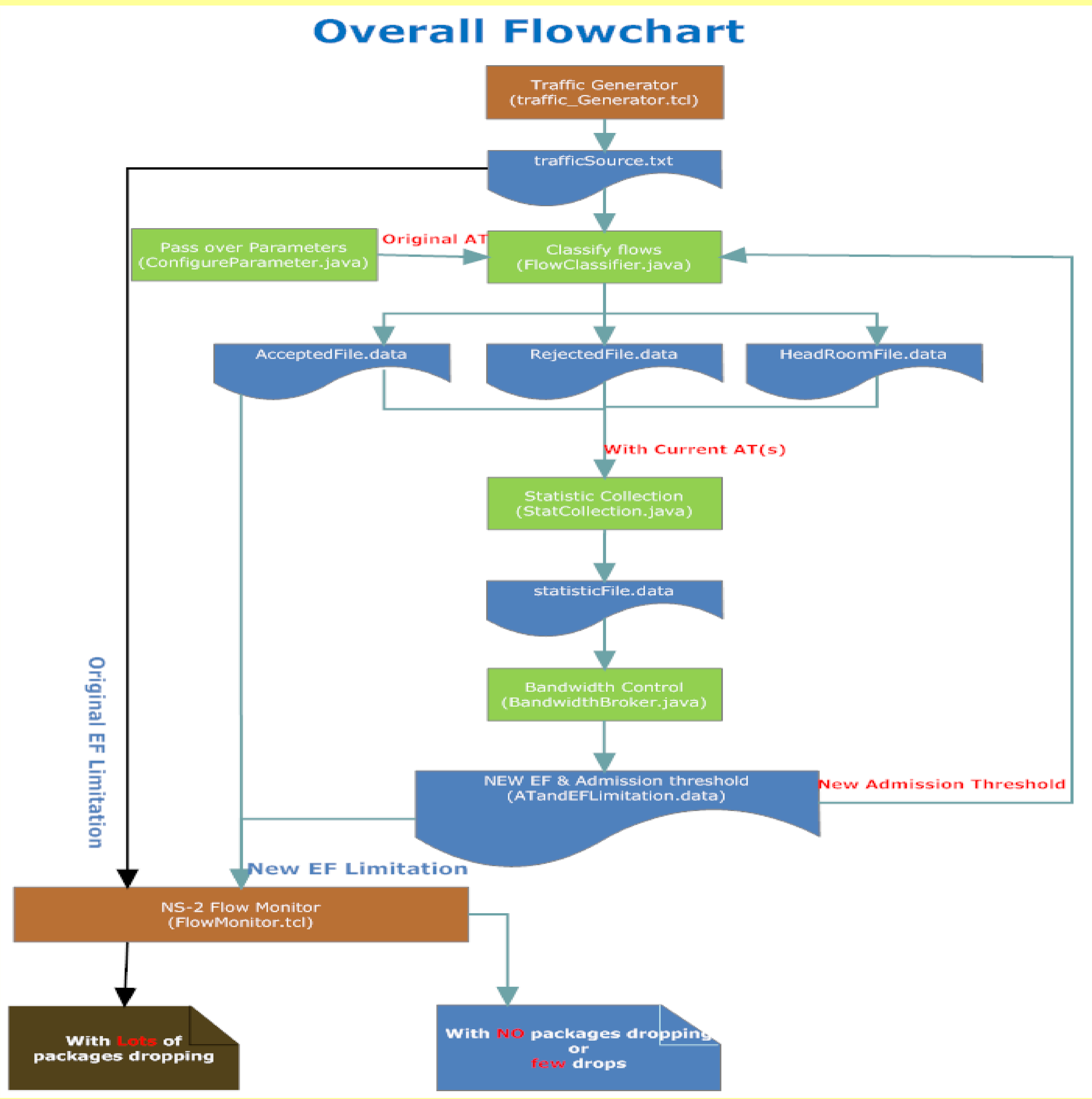
Chenguang Gao, John DeDourek, Przemyslaw Pochec
Faculty of Computer Science, University of New Brunswick Fredericton, NB, Canada

INTRODUCTION

- Multimedia and real-time applications require high quality services.
- Quality of Service (QoS) provides better service such as reduction of the number of dropped packets, delay, jitter, and out-of-order delivery.
- The IETF proposed the Differentiated Services (DiffServ) , which classifies flows with DiffServ code point (DSCP) and the Per-Hop behavior (PHB).
- A Bandwidth Broker (BB) manages the resources based on the Service Level Agreement (SLA) by controlling the network load and by accepting or rejecting bandwidth requests.

OBJECTIVE

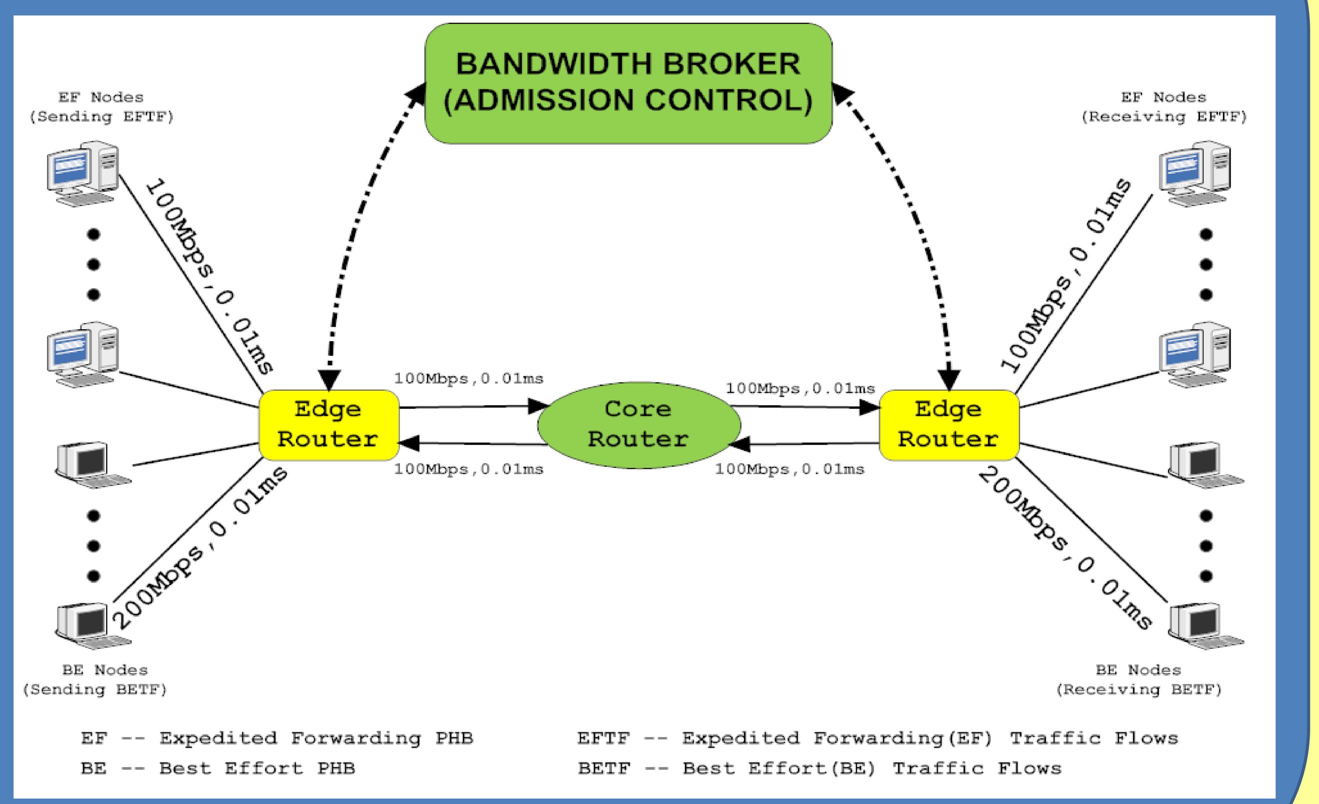
- Design a scheme to provide dynamic bandwidth management in a DiffServ domain with a bandwidth broker.
- Simulate the proposed scheme in NS-2 and analyze results with respect to performance of the admitted streams, and with respect to the cost of unused reserved resources



Methodology

- Flow Generator - randomly generate EF flows and store them into a file
- NS2 DiffServ Script – provide the DiffServ, monitor packet dropping , read incoming flow file and bandwidth file
- Smart Admission Control - read flow files, inspect network load, predict and generate future threshold for EF traffic

Network Topology



Flow Generating Algorithm

```
algorithm --> generating EF flows
for each pair of EF_node($c)
        initialize totol running time (sum($c))
        initialize start_time for each flow (start($c,0))
        initialize stop_time for each flow (stop($c,0))

        set up RNG and its seed (it could be 0 or a constant )
        set up and generate cbr_rate($c,0) with Uniform distribution
        set up and generate idle($c,0) with Uniform distribution
        set up start($c,0) with idle($c,0)
        set up and generate video_length($c,0) with Uniform distribution

        add up $idle($c,0)+ video_length($c,0) to sum($c)
        define stop($c,0) with $sum($c)

        initialize counter d to 1
        while{$sum($c) < 660 seconds}{
                generate rate($c,$d) with Uniform distribution
                generate idle($c,$d) with Uniform distribution
                set up start($c,$d) with [$idle($c,$d) + $stop($c,$d-1)
                generate video_length($c,$d) with Uniform distribution
                add up to $sum($c)+ $idle($c,$d)+ $video_length($c,$d) to sum($c)
                set up stop($c,$d) with the new $sum($c)
                increase counter d
        }
```

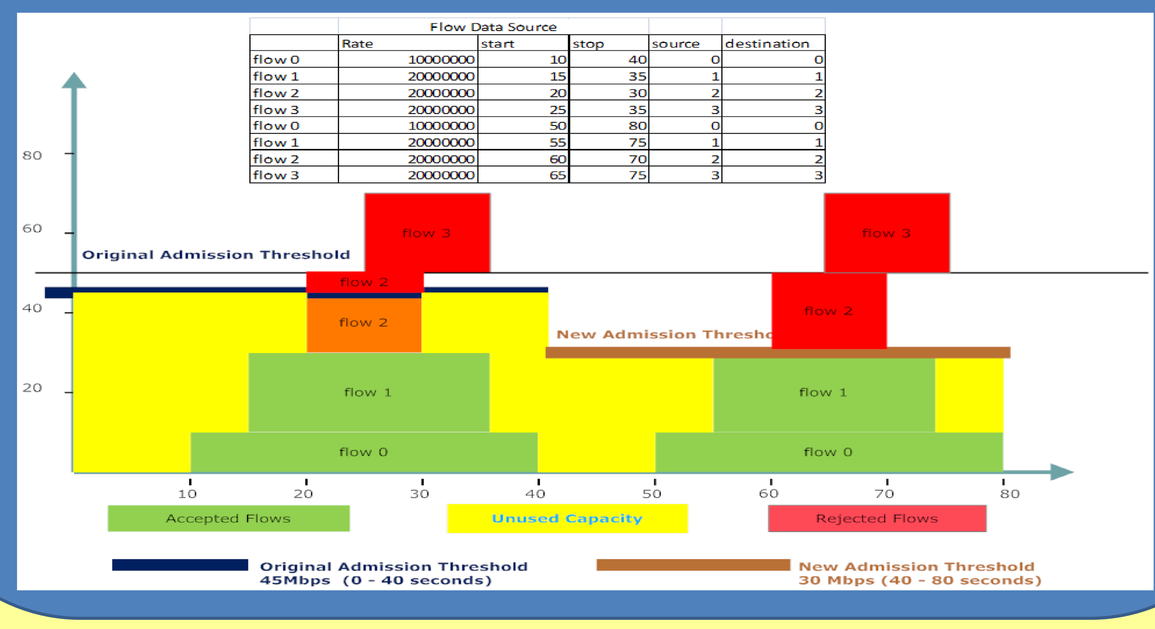
Admission Control Algorithms

- Static Admission Control:

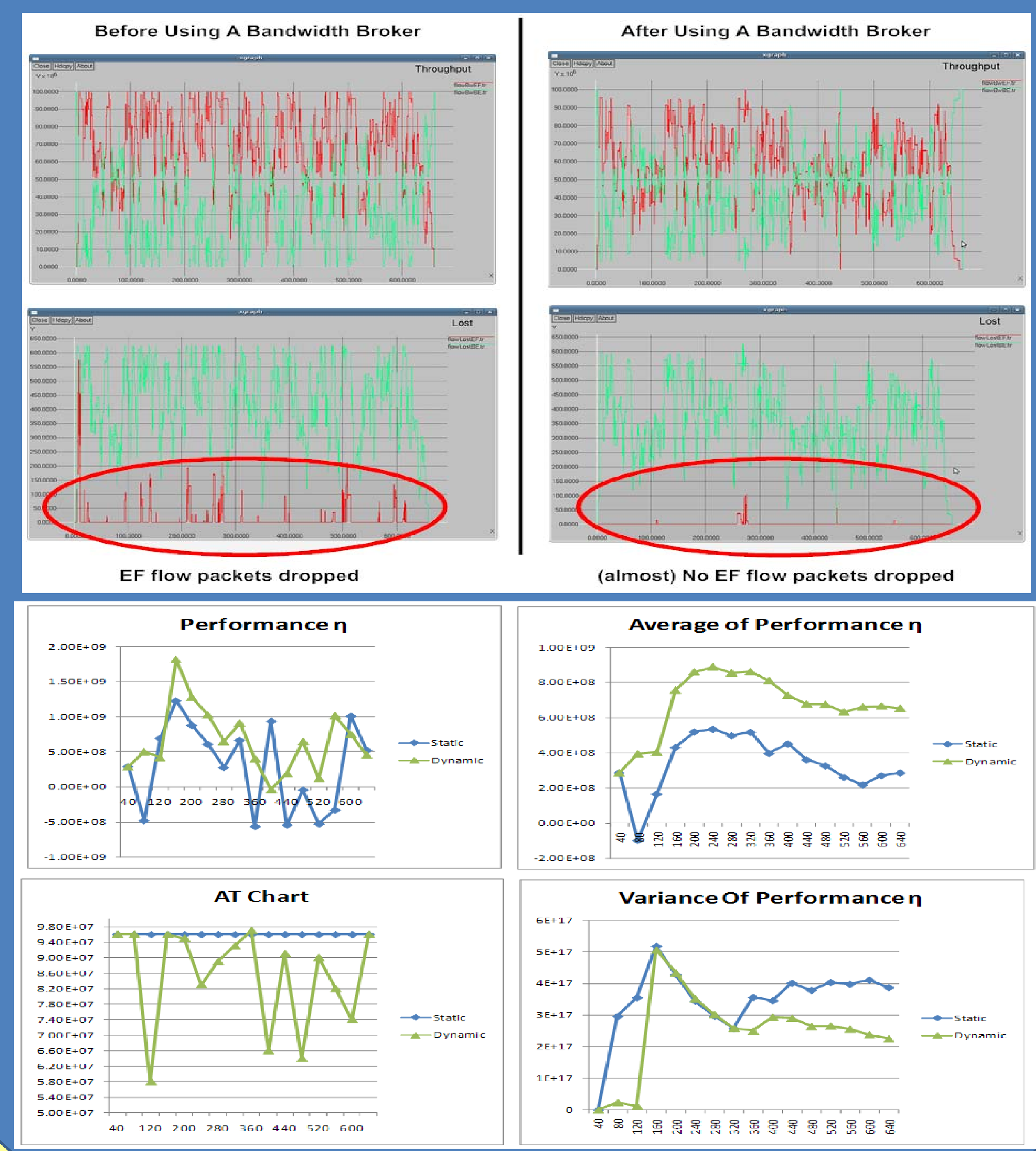
  $AT(0) = AT(1) = AT(2) = \ldots\ldots = AT(N) = \text{Initial } AT$

- Dynamic Admission Control: $AT(N) = AT(N-1)_{(optimal)}$

  $\eta = \alpha \times UsedCapacity - \beta \times UnusedCapacity - \gamma \times RejectFlows \quad (\alpha = \beta = \gamma = 1)$



Experiments and Results



Conclusion

- The simulation shows improved QoS for the EF traffic with dynamic admission control (very few EF packets dropped).
- Performance measured with the metric η is higher for the dynamic algorithm than for the static algorithm.
- The proposed scheme successfully provides dynamic bandwidth management with a bandwidth broker.