ABE: Providing a Low Delay within Best Effort

P. Hurley, M. Kara, J. Le Boudec, and P. Thiran

ICA, Swiss Federal Institute of Technology, Lausanne, Switzerland
Sprint ATL, California
Department of Computer Science, University of Leeds, UK

IEEE Network Magazine
May/June 2001

Introduction

- Multimedia applications can perform well under a wide-range of loss (repair)
- Delay often the major impediment for interactive MM applications
- Internet is "best-effort" with one QoS of traffic for all
  - DiffServ requires monitoring of classes
- Want to keep it simple, but add support for delay sensitive MM traffic
  - → *Alternative Best Effort (ABE)*

Outline

- Introduction **(done)**
- The ABE Service **(next)**
- Implementation
- Simulation Results
- Related Work
- Conclusions

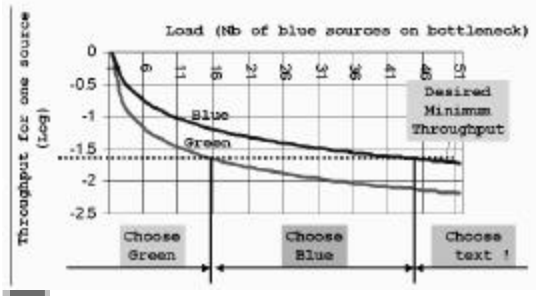
Outline

- Introduction **(done)**
- The ABE Service
  - Definition **(next)**
  - Green does not hurt blue
  - Router requirements
  - Inter-working and Migration
- Implementation
- Simulation Results
- Related Work
- Conclusions

Definition

- ABE packets are either green or blue
  - (Neutral colors, green for "go")
  - Application chooses to make packets green
  - Default is blue
- Green packets get low, bounded delay
- Green does not hurt blue
  - Blue has same or better throughput even if green traffic
- All ABE packets in same best-effort class
  - Traditional congestion control
  - All blue gets more throughput than all green

Possible Packet Coloring Strategy



Assume: utility(rate, delay) = 0 if rate < min
utility(rate, delay) = linear with delay if rate > min

Discussion

- Interactive applications send mix of blue and green
  - "Probe" packets to determine region
- Traditional applications send all blue
  - Care more about throughput
- Note, says nothing about TCP-friendly
  - Still same problem as with best-effort
  - Green makes it no worse since doesn't hurt blue
- Backbones have low delay, so likely ABE in peripheral routers
- Delay bound offered depends upon hops
  - Assume 2-6 low-speed hops
  - Delay 100-150 msec total, maybe 50 for network
  - Per-hop delay about 5-20 msec

Outline

- Introduction **(done)**
- The ABE Service
  - Definition **(done)**
  - Green does not hurt blue **(next)**
  - Router requirements
  - Inter-working and Migration
- Implementation
- Simulation Results
- Related Work
- Conclusions

Green Does Not Hurt Blue

- When there is green traffic in addition to traditional blue traffic, we must have
  - Local transparency to blue
  - Throughput transparency to blue

Local Transparency to Blue

- Consider a traditional router that treated all packets equal (no ABE)
- Should have same delay as traditional router
- If blue not dropped with traditional router, then not dropped with ABE router
- If TCP friendly:

$$\theta = \frac{s}{R\sqrt{\frac{2p}{3}} + 3t_1\sqrt{\frac{3p}{8}}p(1+32p^2)}$$

- What might happen to throughput for green?
  - → Need throughput transparency

Throughput Transparency to Blue

- If green flow is TCP friendly, should get less or equal throughput as blue flows
- Hard to implement exactly since hard to measure
  - Hard to measure TCP friendly, even!
  - Consider it to be a loose requirement
- Implement by making sure green has higher loss ratio

Outline

- Introduction **(done)**
- The ABE Service
  - Definition **(done)**
  - Green does not hurt blue **(done)**
  - Router requirements **(next)**
  - Inter-working and Migration
- Implementation
- Simulation Results
- Related Work
- Conclusions

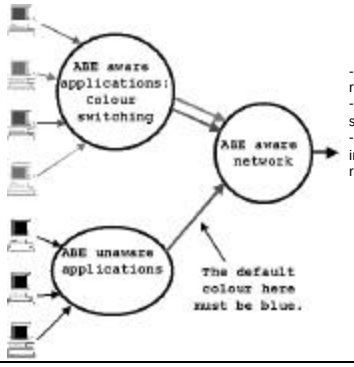
Router Requirements

- Provide low, bounded delay to green
- Provide local transparency to blue
- Provide throughput transparency to blue
- Preserve packet sequence within blue and green
  - May be out of order across colors
- Keep green packet loss as low as possible
  - Make green attractive as possible

Outline

- Introduction **(done)**
- The ABE Service
  - Definition **(done)**
  - Green does not hurt blue **(done)**
  - Router requirements **(done)**
  - Inter-working and Migration **(next)**
- Implementation
- Simulation Results
- Related Work
- Conclusions

Interworking and Migration



- Can add one router at a time
- Let customers switch to gradually
- Should not impact other routers

Outline

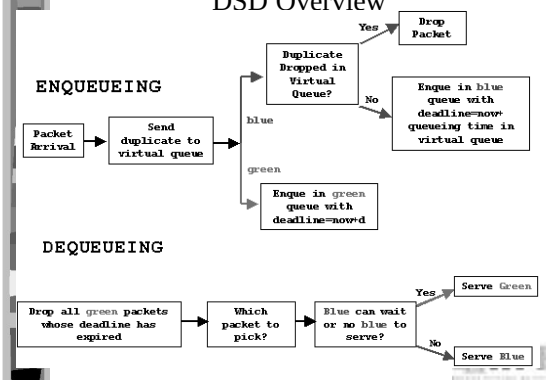
- Introduction **(done)**
- The ABE Service **(done)**
- Implementation **(next)**
  - Duplicate Scheduling with Deadlines
  - Properties of (DSD)
- Simulation Results
- Related Work
- Conclusions

Implementation

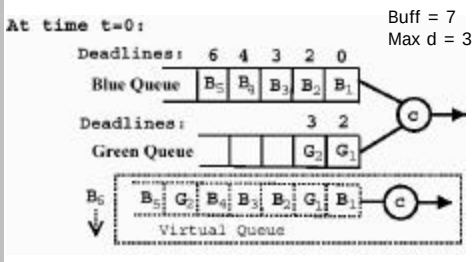
- Could try modified FCFS:
  - For blue, enqueue normally
  - For green, drop if delay > max
  - (What is a problem with this?)
- Instead, use separate queues
  - But still *work conserving*
- Deadlines associated with each packet
  - Dequeue color that has earlier deadline
  - If both, use a control function for fairness

→ Duplicate Scheduling with Deadlines (DSD)

DSD Overview

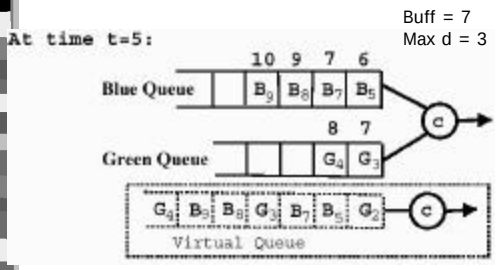


DSD Example



Serve: G1, B2, B3, B4
Drop: G2 (deadline missed), B6 (buffer full)

Duplicate Scheduling with Deadlines



Serve: G3, B5, B7, 4g, B8 and B9

DSD Modifications

- Only enqueue green packet if length of green queue + blue packets with deadline less than $d < d$
  - So, would not have enqueued G2
- If either can be served, if [0,1] < g then pick green else blue
  - g=1, favor green, g=0 favor blue
  - (g=1 in example)
- Can also use active queue management (AQM) for congestion monitoring

Properties of DSD

- Buffer always less than *Buff* because of virtual queue
- All blue packets served by deadlines, so same as or earlier than best-effort
- All green packets served before *d*, else dropped

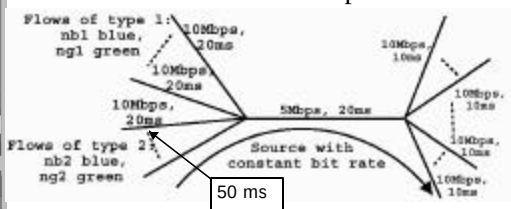
Outline

- Introduction **(done)**
- The ABE Service **(done)**
- Implementation **(done)**
- Simulation Results **(next)**
- Related Work
- Conclusions

Simulation

- Done in NS-2
- Show green does not hurt blue
- Show green benefits from low delay
- Show loss rates for both types
- Compare to reference condition, flat best-effort FCFS (droptail) router

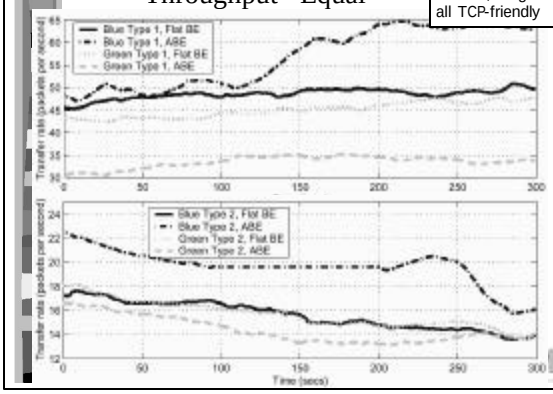
Simulation Setup



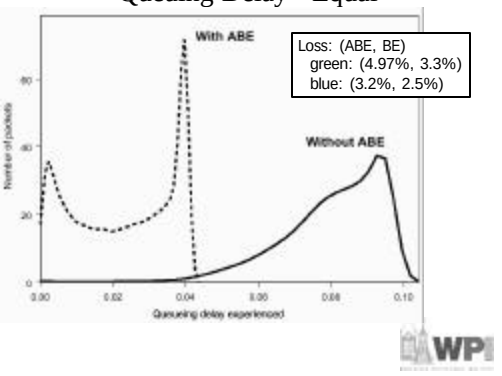
- blue are TCP-Reno, green are TCP-Friendly [BB00]
- Some simulations have one additional green source that is unresponsive CBR
- packet size 1000 bytes
- delay max = 0.04 seconds
- simulations run for 300 seconds

Throughput - Equal

10 blue, 10 green all TCP-friendly

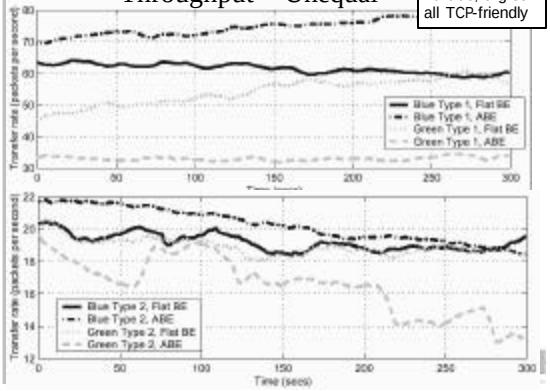


Queuing Delay - Equal



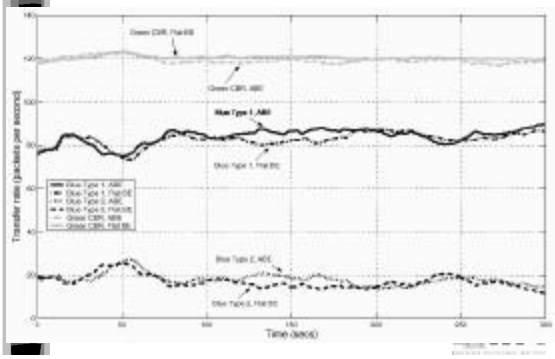
Throughput - Unequal

10 blue, 6 green all TCP-friendly



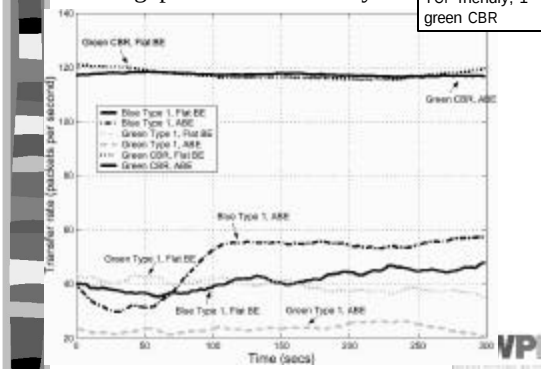
Throughput – CBR

10 blue, 1 green CBR



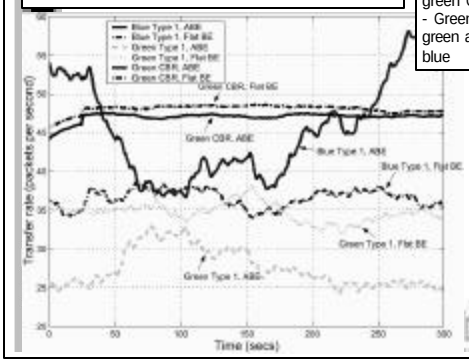
Throughput – CBR + Friendly

10 blue, 10 green TCP-friendly, 1 green CBR



Throughput – Mixed Green + Blue

- 10 blue, 10 green TCP-friendly, 1 green CBR
- Green does 80% green and 20% blue



Outline

- Introduction **(done)**
- The ABE Service **(done)**
- Implementation **(done)**
- Simulation Results **(done)**
- Related Work **(next)**
- Conclusions

Related Work

- IntServ
  - admission control plus reservation
  - Per-flow accounting and charging
  - Doesn't scale
  - May perform on edge only
- DiffServ
  - Aggregates (classes) of flows
  - Scales better

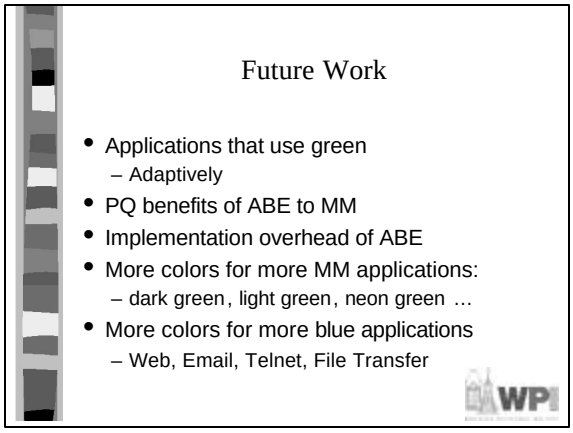
Related Work

- Low delay service
  - Crowcroft et al (also gets more throughput)
  - EF provides low delay and low loss
  - SIMA has level for how 'real-time' traffic is
- Low delay class
  - Dovrolis et al
  - AF – Assured Forwarding
- All require changes to existing price structures. Incremental deployment difficult.

Conclusion

- ABE
  - Supports low delay
  - No reservation or signaling required
- Choice of green or blue up to application
- One ABE implementation presented (DSD)
- Simulation and implementation suggest:
  - Green benefits from lower delay
  - Blue not harmed
  - Under a variety of conditions

Future Work?


Future Work
• Applications that use green
– Adaptively
• PQ benefits of ABE to MM
• Implementation overhead of ABE
• More colors for more MM applications:
– dark green, light green, neon green …
• More colors for more blue applications
– Web, Email, Telnet, File Transfer
WPI