Passive Aggressive Measurements with MGRP

Pavlos Papageorge, Justin McCann and Michael Hicks
ACM SIGCOMM 2009
{University of Maryland, College Park}

Vaibhav Bajpai
NDS Seminar 2011

Outline

- Introduction
- Objective
- Approach
- Implementation
- Performance Evaluations

What is Network Measurement?

- is a process of collecting data that measure certain phenomena about the network
  - should be a science
  - today: closer to an art form
- bread and butter of networking research
  - deceptively complex
  - probably one of the most difficult things to do correctly

Why Measure Network Traffic?

- need to adapt to changing network conditions
- optimal overlay construction
  - peer-to-peer communication
  - end-host multicast
  - secure overlay services (SOS): proactive DoS prevention
- optimal service selection
- multi-path routing

Known Techniques

- Active Network Measurements
  - actively inject probe packets to see how network responds
- Passive Network Measurements
  - passively observe existing traffic

Passive Measurements: UseCases

- dynamic stream switching by analyzing rate of buffer use [adobe flash media server].
- reducing stream rate by monitoring packet and frame loss [skype]
- [+] low overhead

Passive Measurements

Problems

- inadequate to detect when conditions improve
- are not flexible/modular
  - tightly coupled with application using them: TCP/RTCP
- estimation is slower and less accurate
  - lack of control over the probe sequence

Active Measurements: UseCases

- periodically improve the quality: hoping traffic can be supported
- using specific tools:
  - pathload, pathchirp: (probe available bandwidth)
  - badabing: (loss estimation)
- application shaping its own data as measurement tool
- [+] fairly accurate!

Active Measurements

Problems

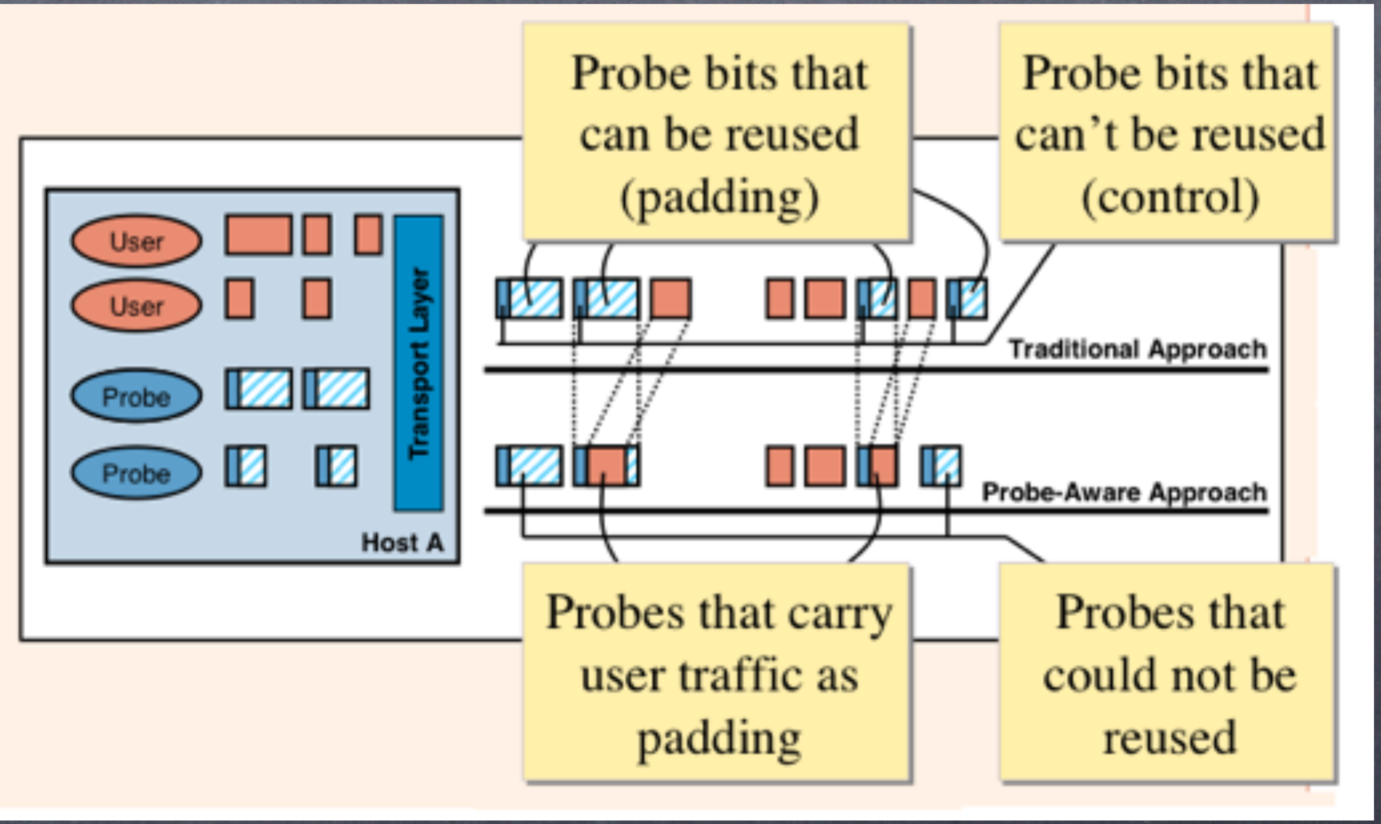
- tools steal bandwidth away from user data
  - just ping traffic on planetLAB averages around 1GB/day
- prohibitive during conditions of congestion

Objective

"minimize the bandwidth that measurement tools consume while maintaining the same level of accuracy and timeliness"

Approach

- probe packets are currently treated in same way as user packets; but probes consist mostly of empty padding bits.
- there exists an opportunity to reuse the empty padding bits to carry user data!



Outline

- Introduction
- Objective
- Approach
- Implementation
- Performance Evaluations

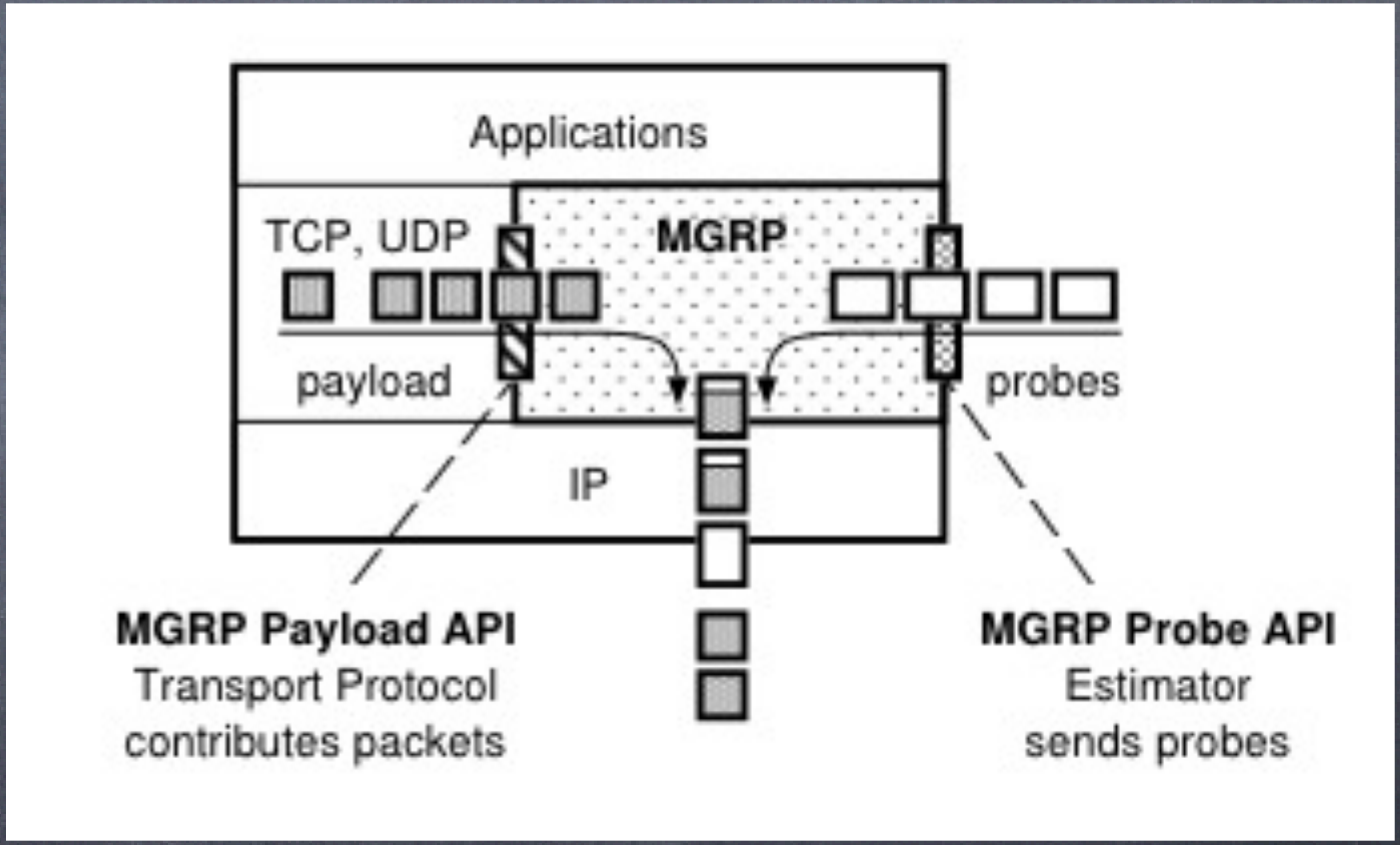
MGRP

Measurement Manager Protocol

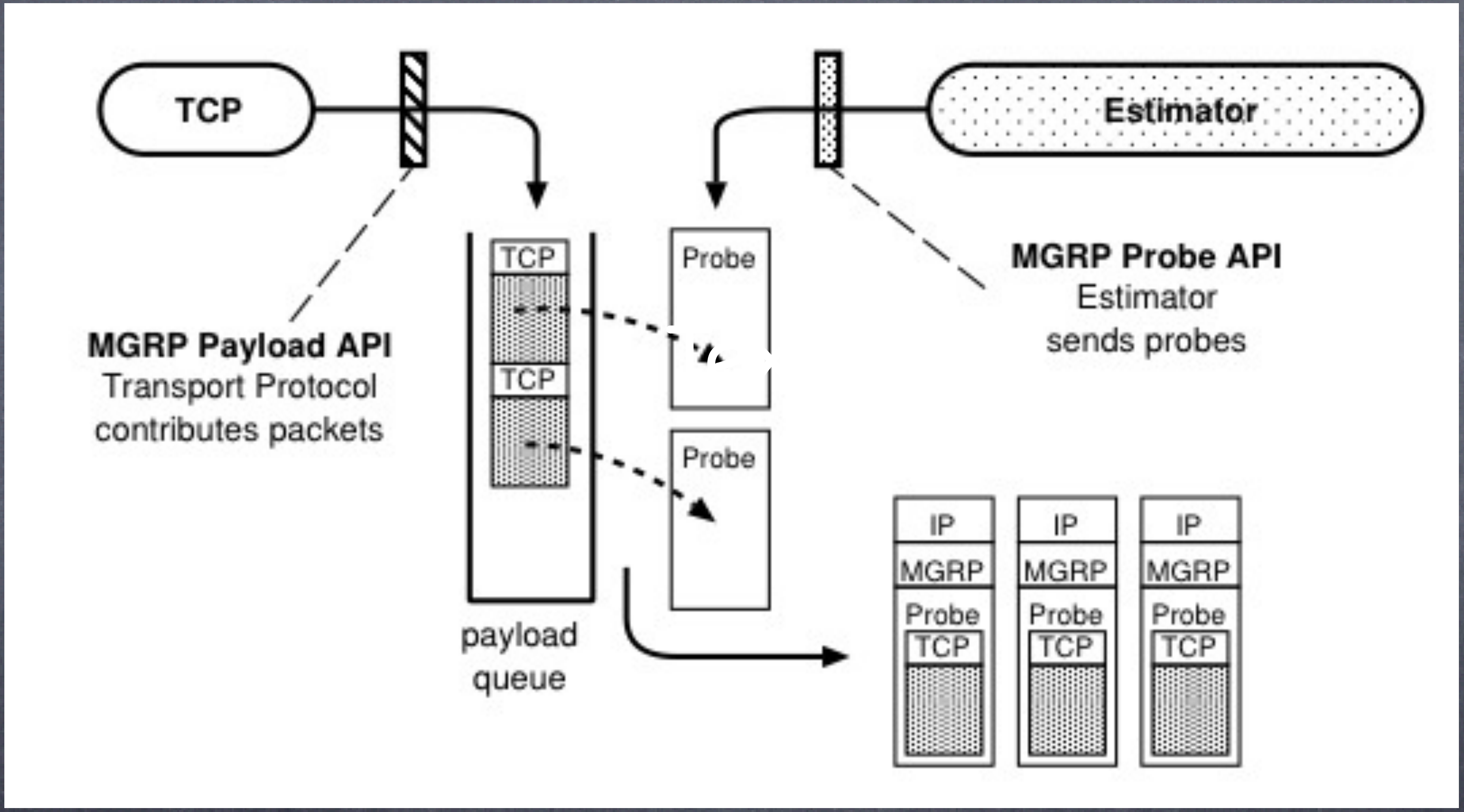
- in-kernel implementation sitting at Layer4 alongside a modified variant of TCP.
- transparently piggybacks user data inside probes.
  - active measurements tools send probe informations to MGRP using a probe API: number and size of probes, amount of padding, gap between probes et al.
- multiplexes all flows into a single stream.
  - allows unified congestion control across all participating flows
- schedules user data for maximal use of padding.

MGRP APIs

- payload API
  - intended for transport protocols to send user data
- probe API
  - intended for active tools to send probes



MGRP
payload queues: maximize padding



lower the chances of a probe going out without a transport payload, at the cost of a slightly increased RTT

MGRP
Probe API

- probe transaction: group of probes that need to be sent out according to a particular sending pattern.
  - the first probe of the group may be delayed but once it goes, the whole group has to go.
  - estimators raise the barrier in MGRP, send all the packet probes and finally lower the barrier.
- probes are sent using `sendmsg()` and received using `recvmsg()`
- received probes contain: header + ancillary data: (sender + receiver) timestamps

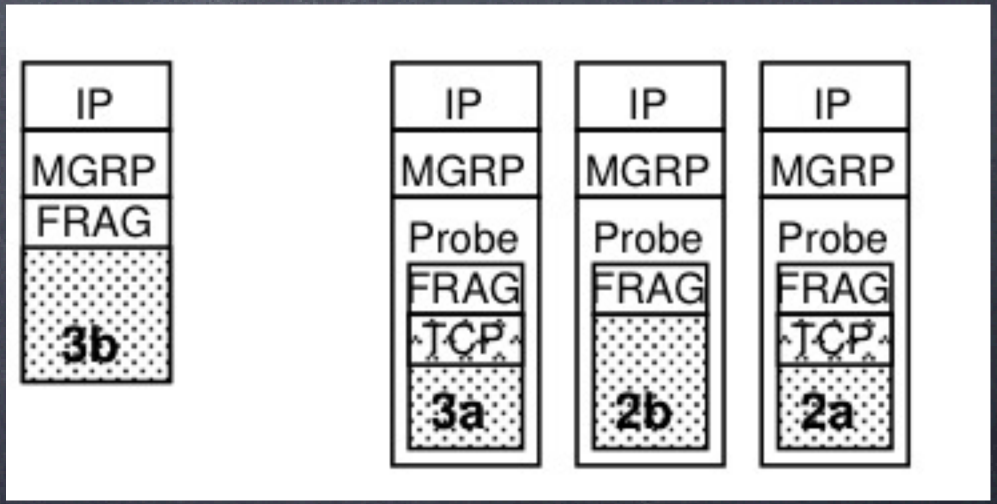
MGRP

Payload API

- transport protocols treat MGRP like the IP layer: stick packets in and expect to pop at remote end.
- issue: transport protocols typically segment to fit the MTU
  - fragment the segments at MGRP; much like IP fragmentation.

or

  - transport protocols themselves issue instructions on how to segment into small packets.



MGRP
Benefits of Kernel-Based Implementation?

- can piggyback data from any application.
- applications need no modification, measurement tools need modest change.
- inter-probe gaps have high precision and low overhead using high-resolution kernel timers!

MGRP
Effects on Application Data

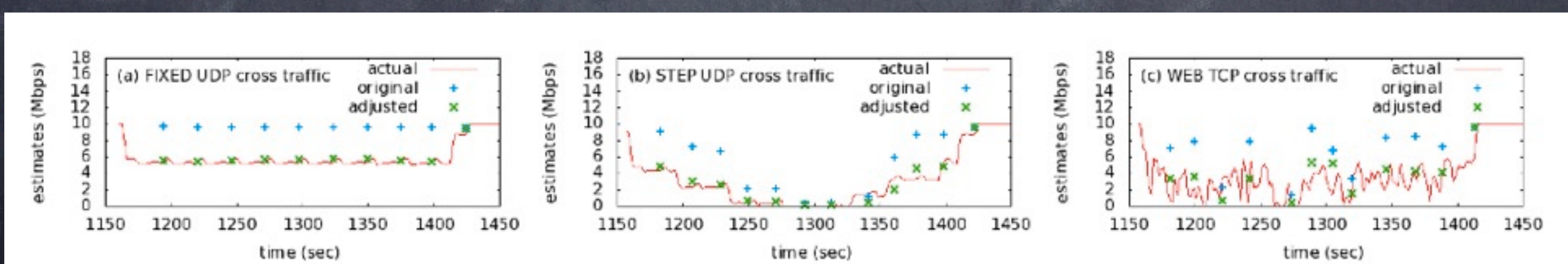
- piggybacking increases the chance that a lost packet is a user data.
  - now sent at probe burst-rate of high instantaneous bandwidth, increasing likeliness of loss.
- design decisions
  - on buffer size? (increased piggyback vs delayed TCP response to packet loss)
  - which probes to piggyback on? (packets at end of burst more likely to be dropped on congestion)

MGRP
Effects on Measurement Tools

- advantages:
  - can send probes more aggressively.
  - converges more quickly.
  - provides more accurate results.
- issues?
  - on significant piggyback, overestimates available b/w.

MGRP
Effects on Measurement Tools

- issues?
  - on significant piggyback, overestimates available b/w.
- solution?
  - query MGRP via an `ioctl` call to learn piggybacking characteristics of the recent transaction to adjust estimates



MediaNet

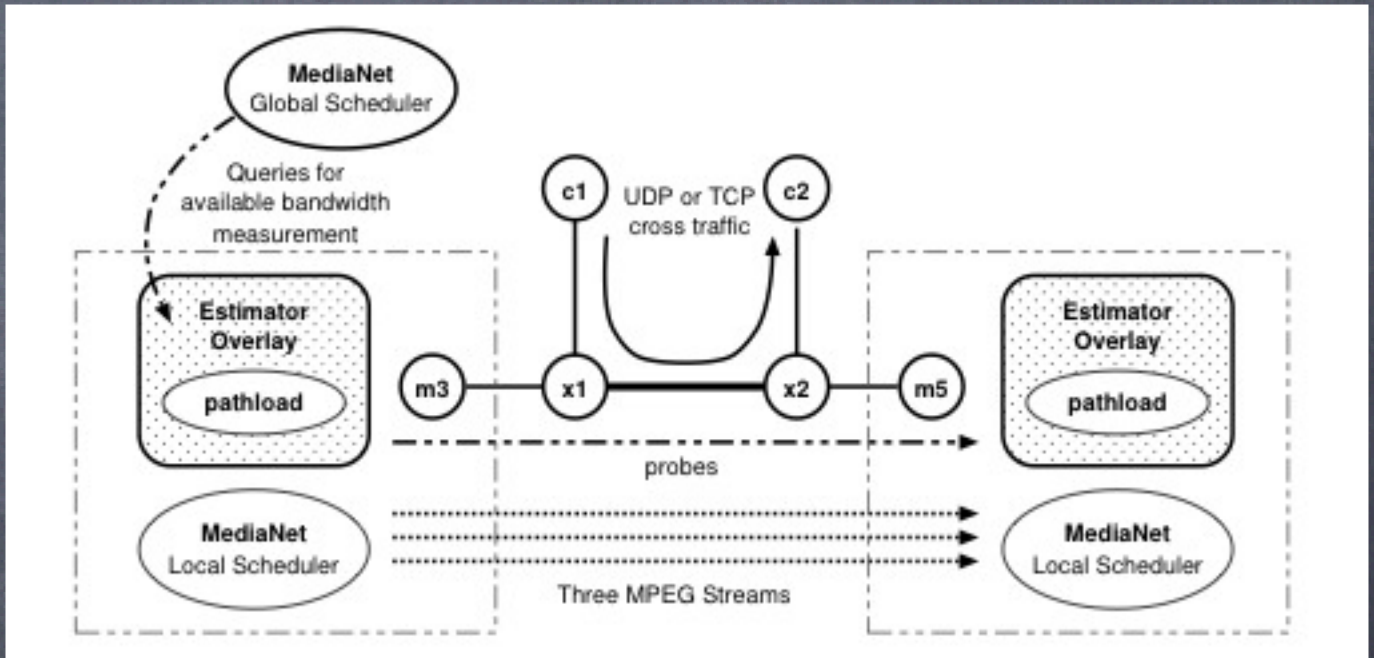
- overlay to provide adaptive, user-specified QoS guarantees for media streams
- local schedulers: apply adaptations while forwarding traffic
- global schedulers: chooses a delivery path for a stream; deploys LS specific adaptations
- problem? uses purely passive measurements
  - GS cannot tell when additional b/w is available!

Measurement Overlay

- user-space implementation to measure available bandwidth
- actively measures its virtual links
- applications query the overlay to acquire up-to-date path conditions

Measurement Overlay

- GS modified to periodically query overlay
- LS needs no modifications
- GS can now safely increase streaming rates when MO reports higher available bandwidth



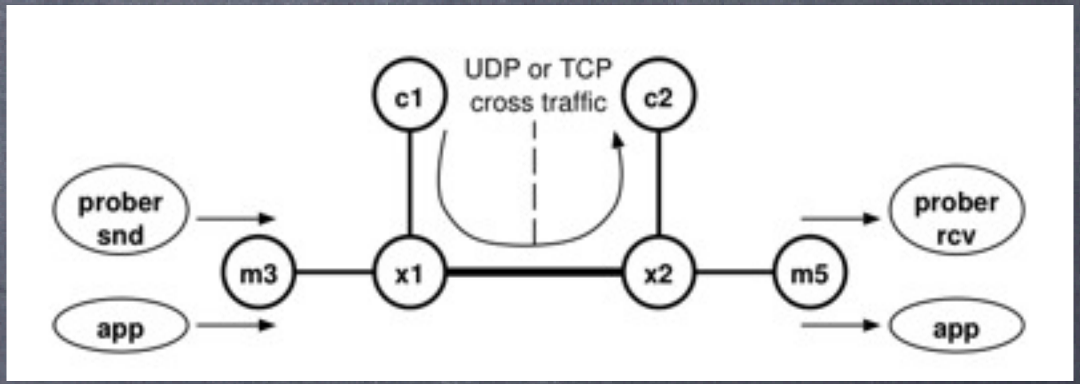
Outline

- Introduction
- Objective
- Approach
- Implementation
- Performance Evaluations

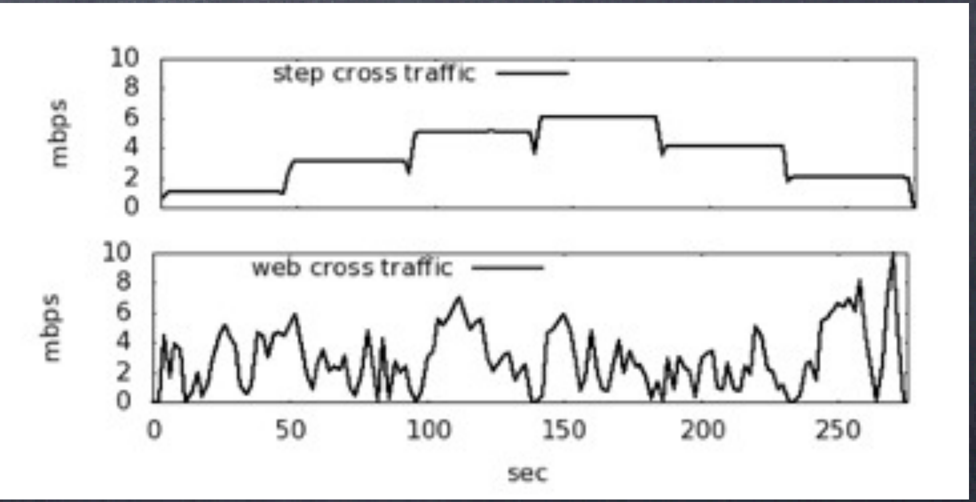
MGRP

Experimental Setup

- Source Traffic: (emulation using nuttcp, 4Mbps constant)
- Probe Traffic:
  - pathload: (fluctuates amount of probe traffic)
    - pSlow: pause b/w trains = RTT + 9*TX
    - pFast: pause b/w trains = 1 RTT
  - synthetic: (oblivious to change in n/w state)

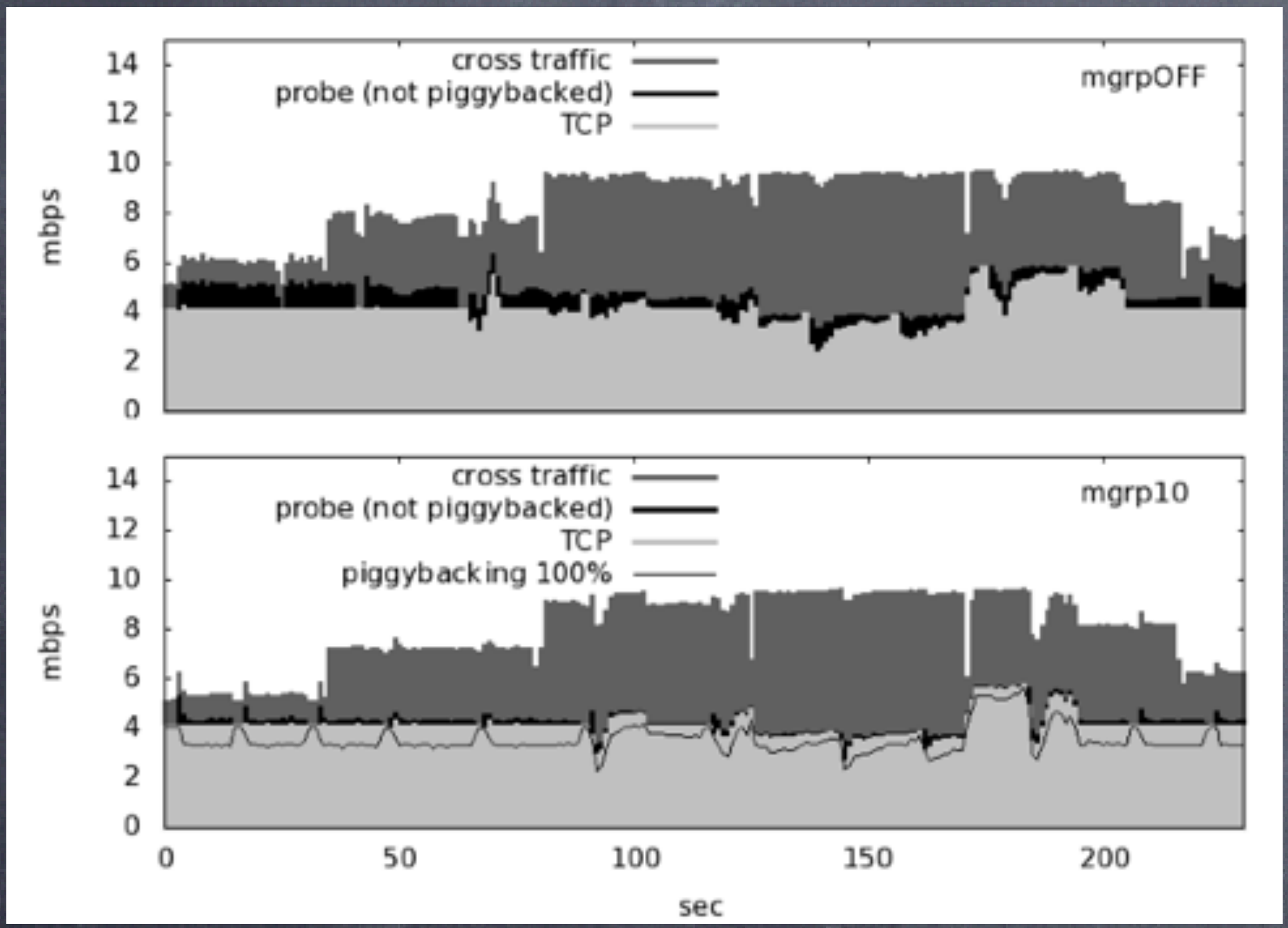


- Cross Traffic:
  - STEP: stepwise UDP using tcpreplay
  - WEB: poisson distributed TCP using NTools

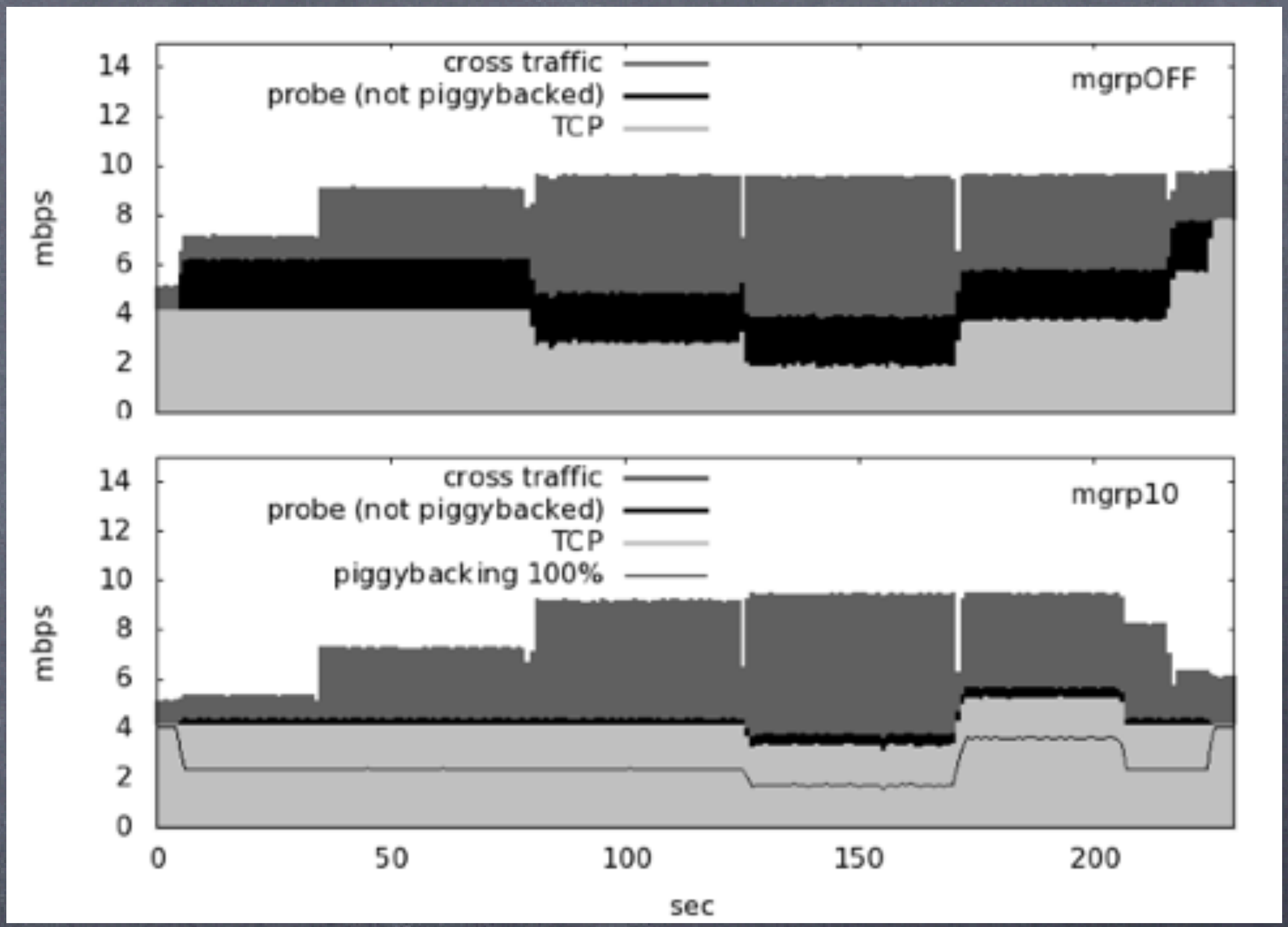


MGRP

STEP: Results



pFast probes

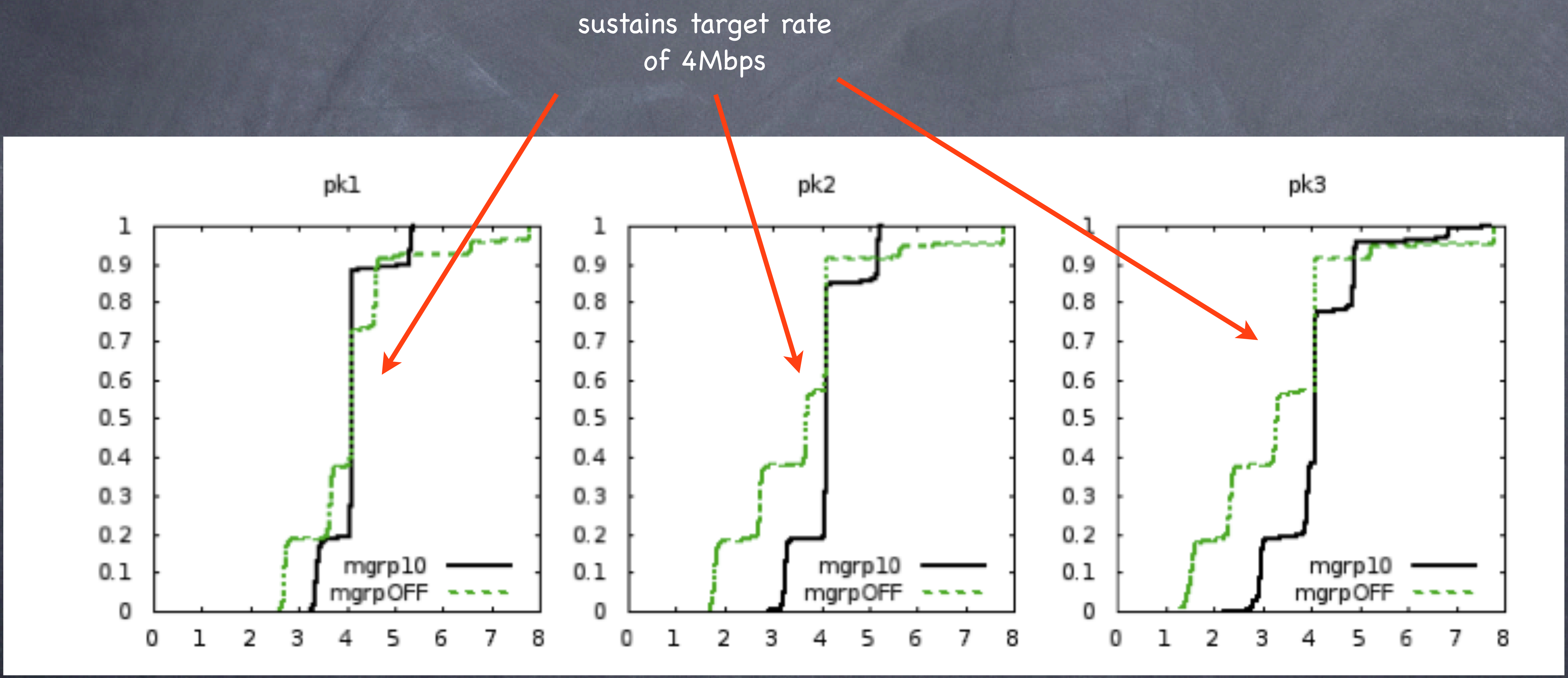


synthetic probes

significant piggyback: nearly eliminating probing overhead.

MGRP

STEP: Results



CDFs of source throughputs while running synthetic probe trains

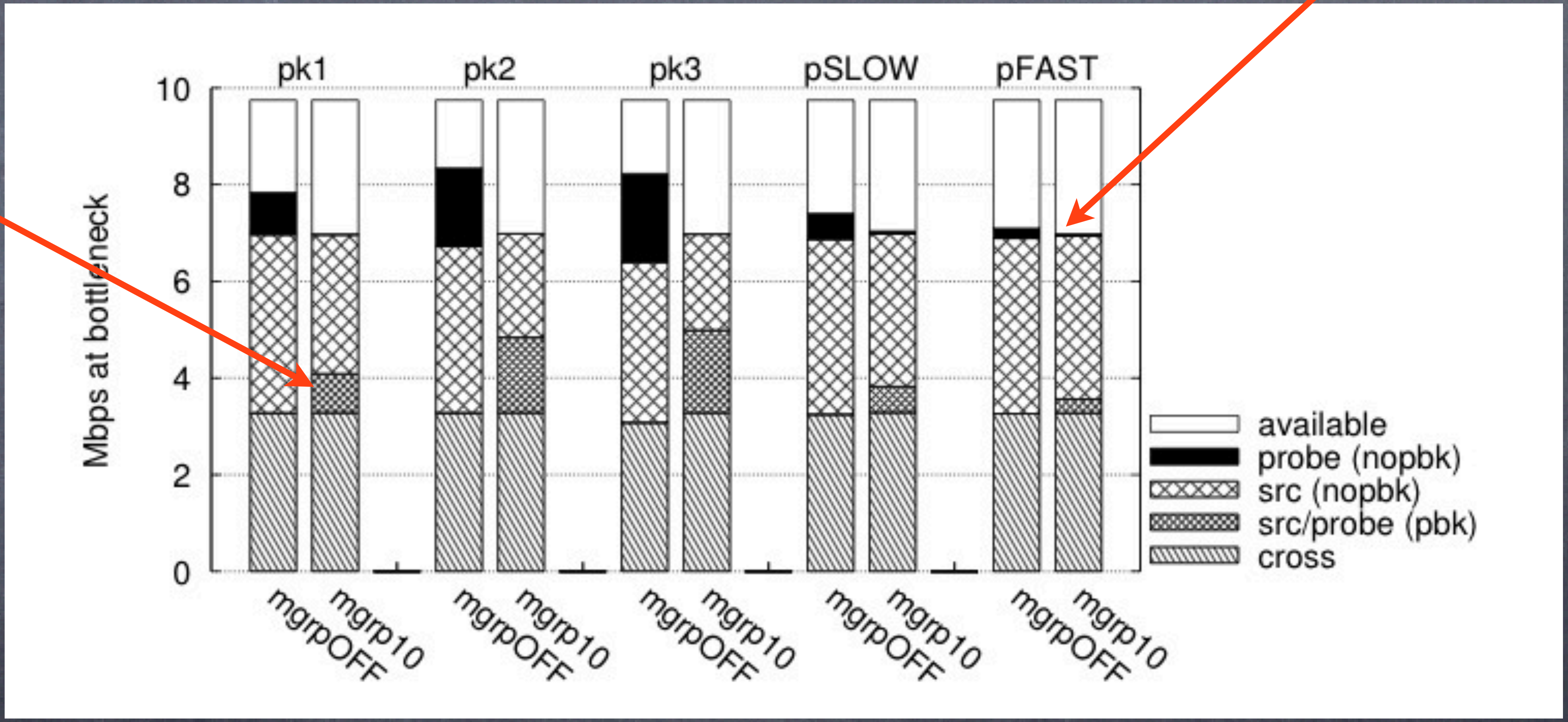
smoothens out source traffic: competes less with probes

MGRP

STEP: Results

minimal probes with no piggyback

riders contribute to source traffic



cross-traffic sustains requested rates

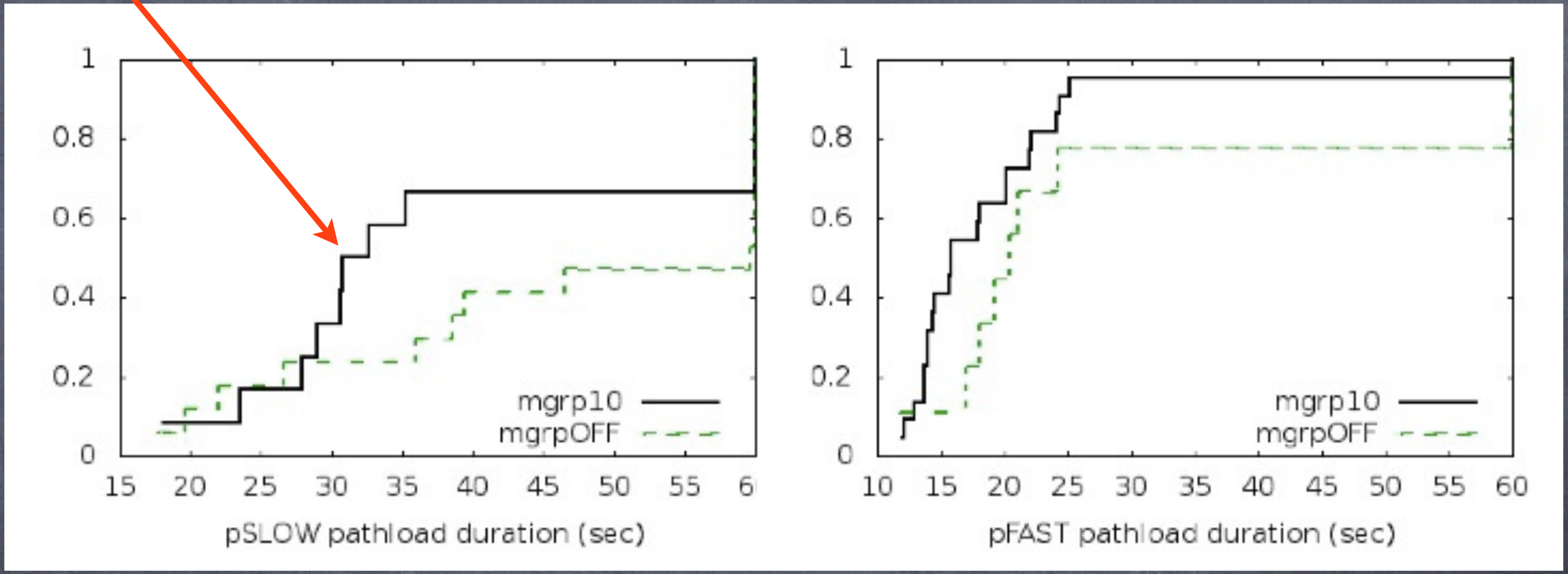
no adverse effect on UDP cross traffic

MGRP

STEP: Results

with MGRP: 50% reached in no time!

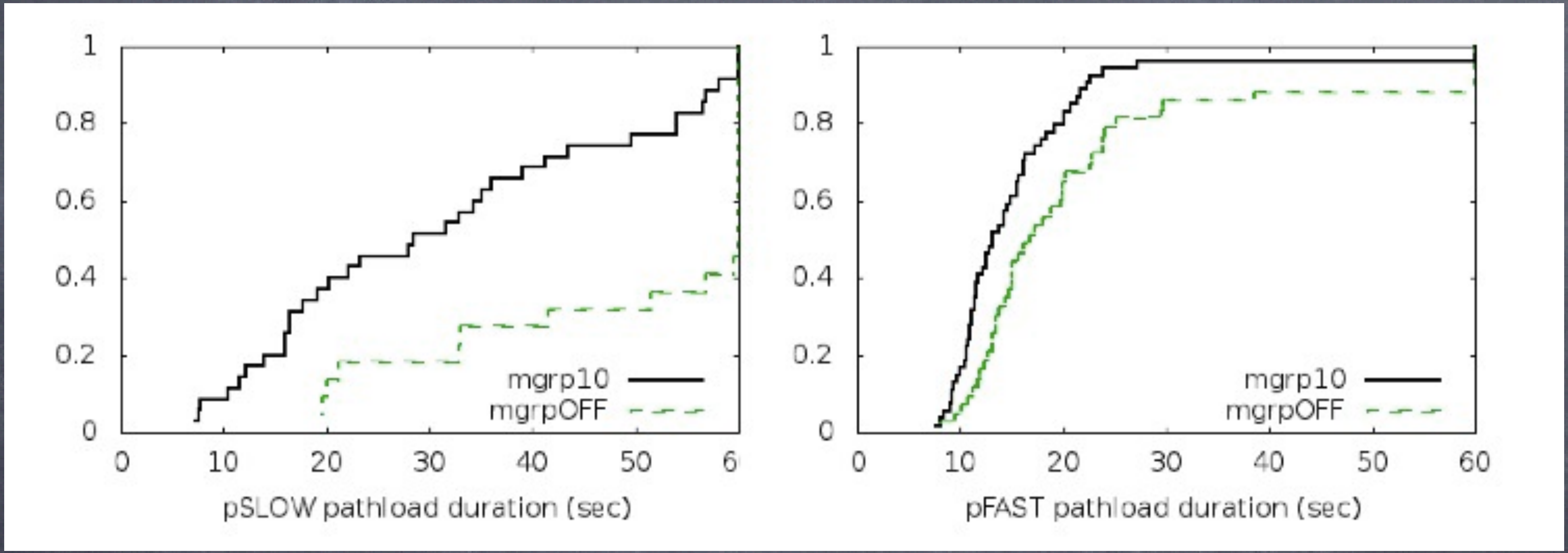
without MGRP: 48% fail to complete at all



CDFs of pathload completion times

pathloads complete their measurements more quickly

WEB: Results



CDFs of pathload completion times

pathloads complete their measurements more quickly

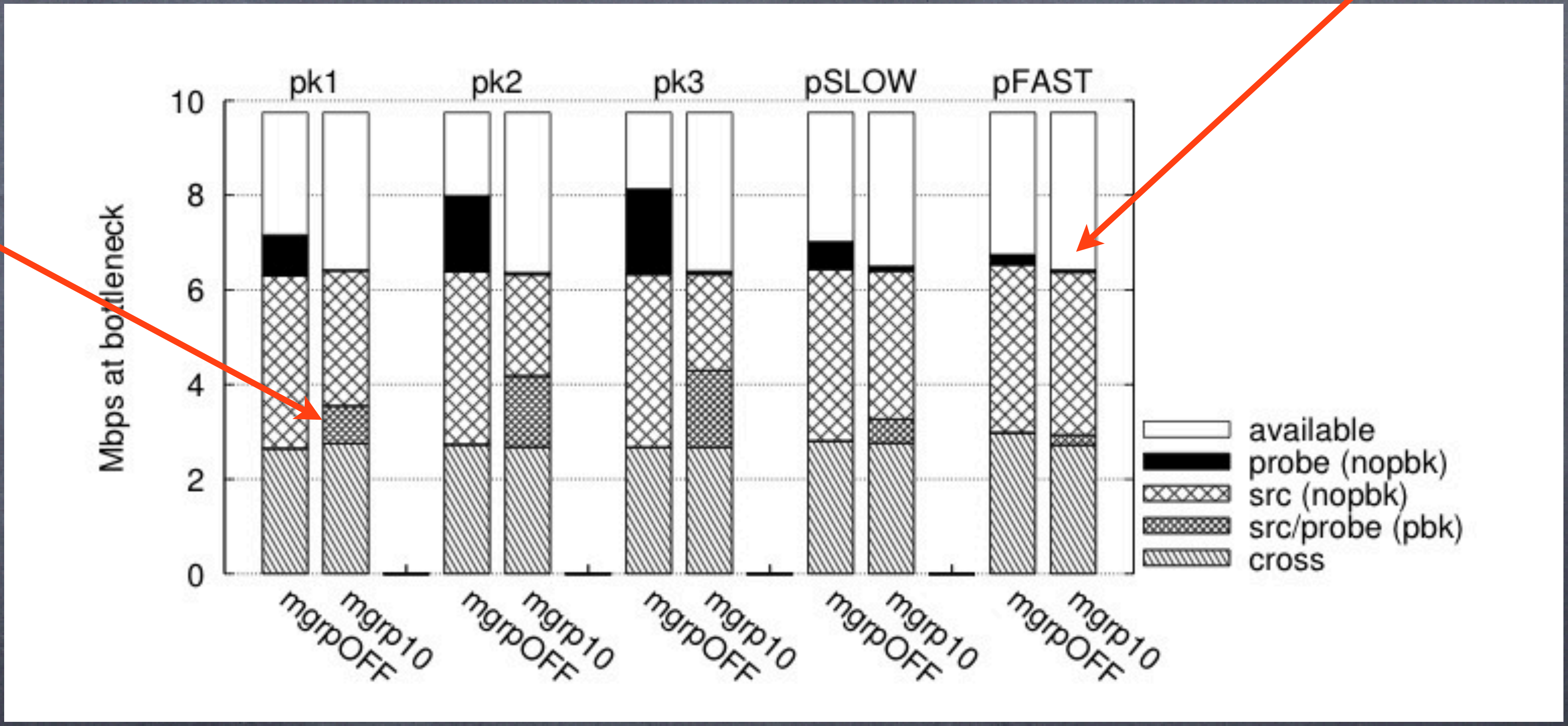
very similar to STEP results

MGRP
WEB: Results

minimal probes with no piggyback

riders contribute to source traffic

cross-traffic sustains requested rates



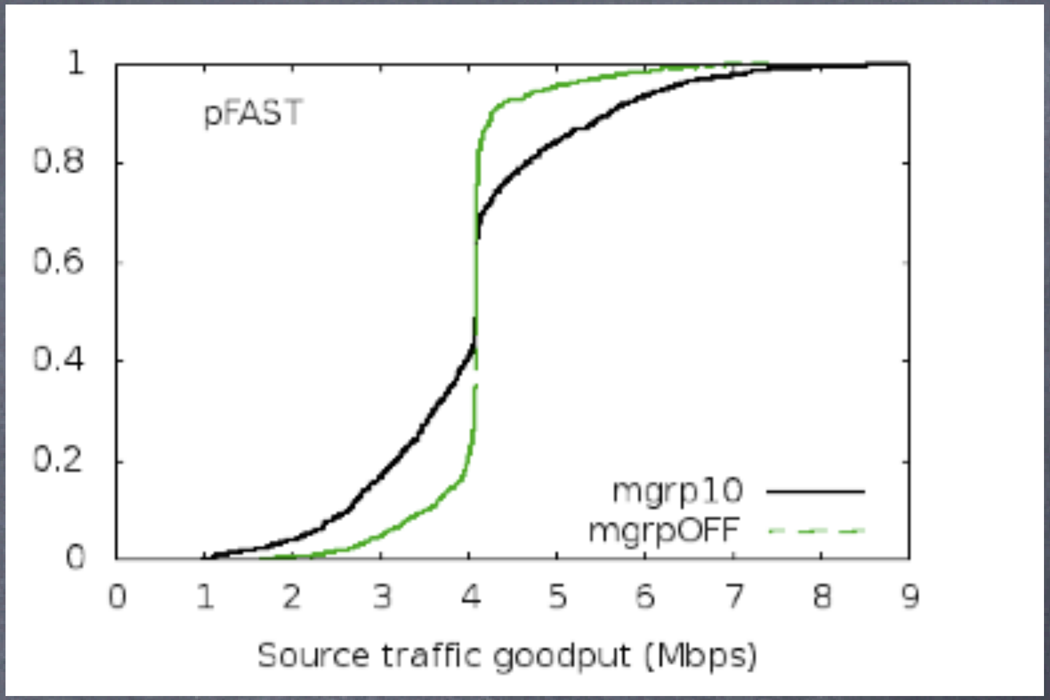
almost no adverse effect on TCP cross traffic

very similar to STEP results

MGRP

WEB: Results

- problem?
  - web cross traffic is highly variable
  - pathload is adaptive



fails to sustain target rate of 4Mbps

- solution?
  - selectively piggyback only on first portion of high rate trains
  - policy tuning knob to control maximum % of riders in a train
  - remove the shared fate problem

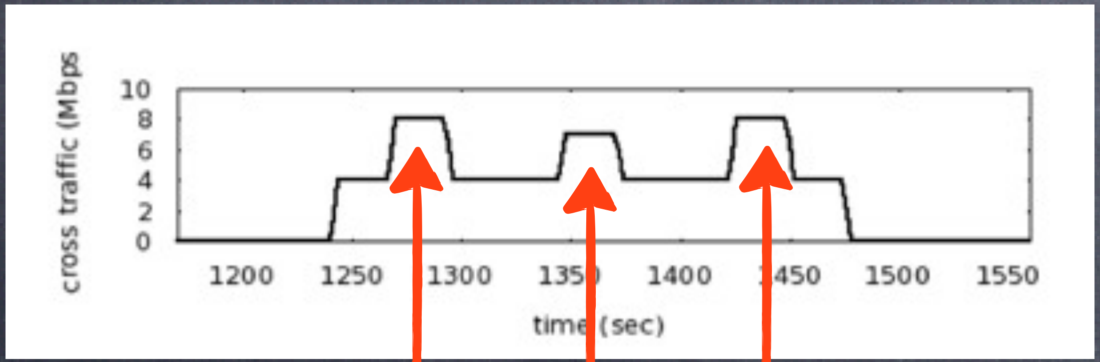
Measurement Overlay

Experimental Setup

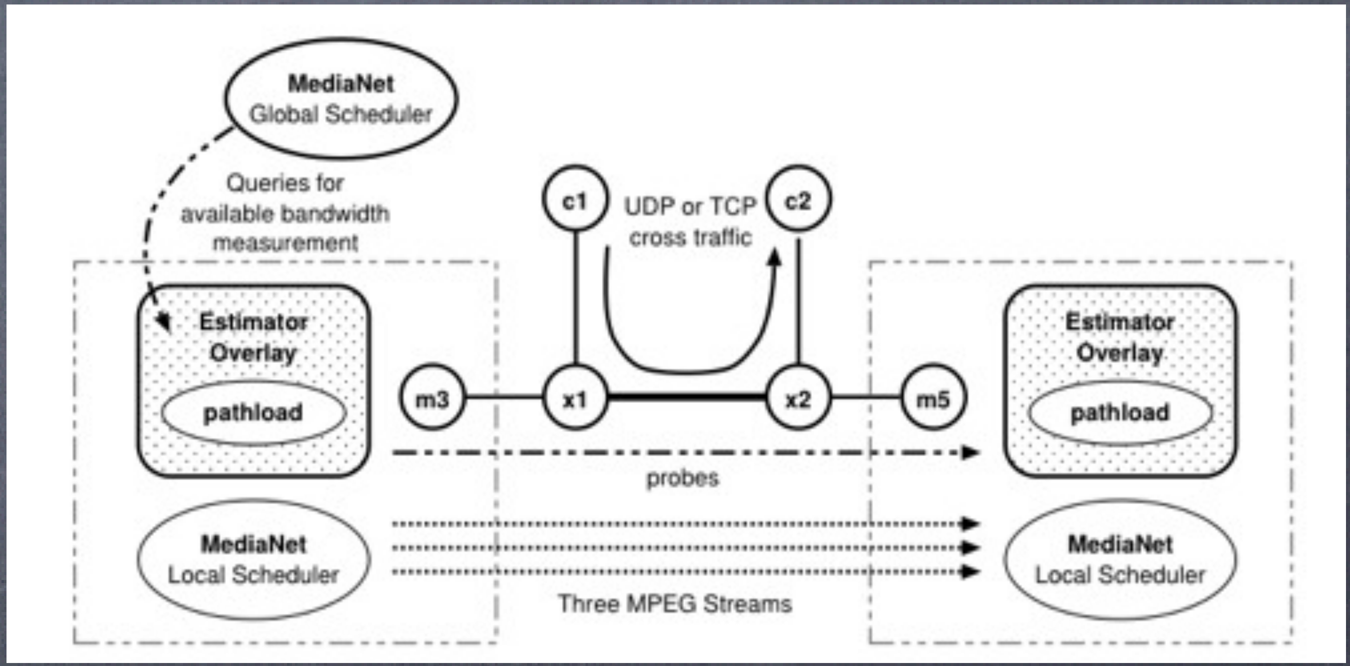
- Source Traffic (adaptations take place by dropping frames.)

Frame Type	Average Size (B)	Frequency (Hz)	Add'l BW (Kbps)
I	13500	2	216
P	7625	8	488
B	2850	20	456

- Cross Traffic:



less opportunity to increase transmission rate



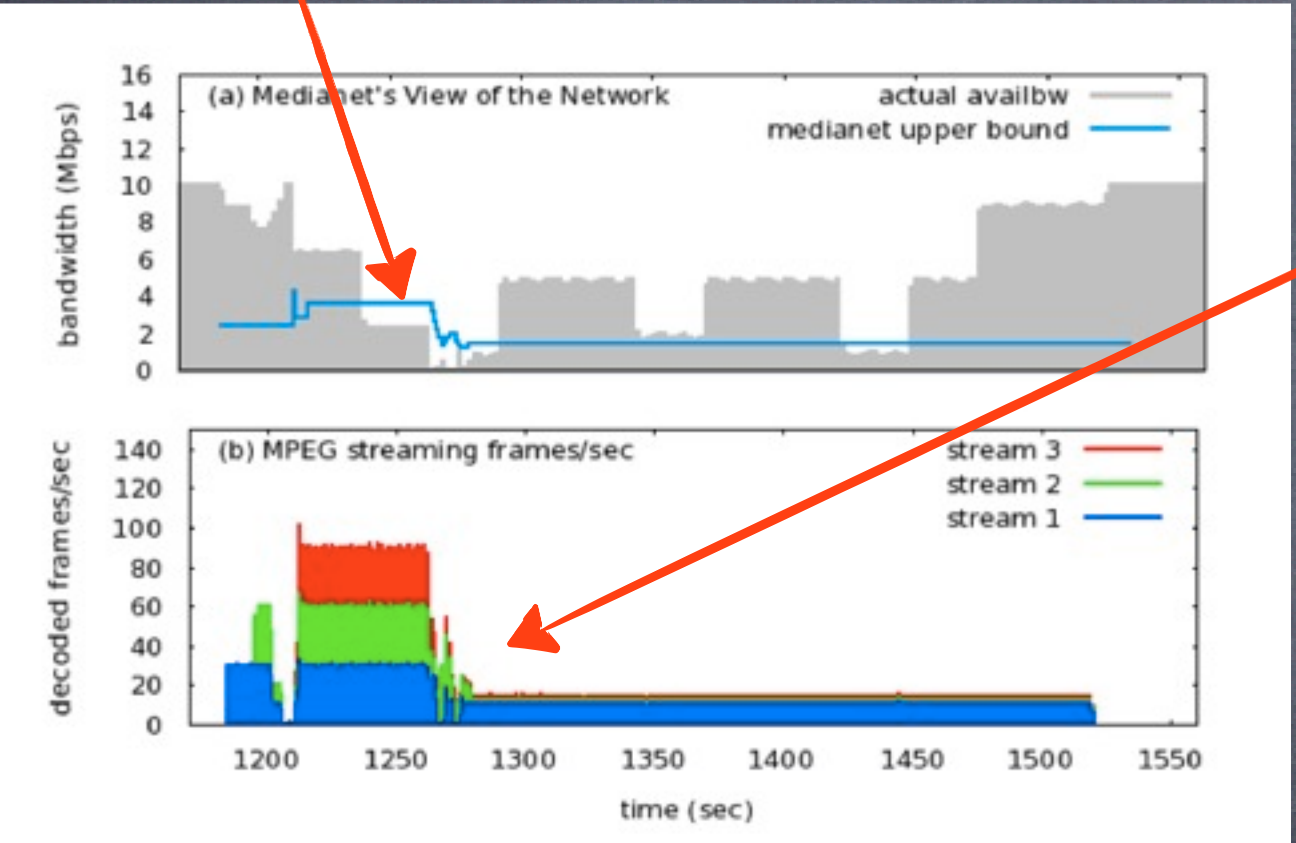
Measurement Overlay

MediaNet Results

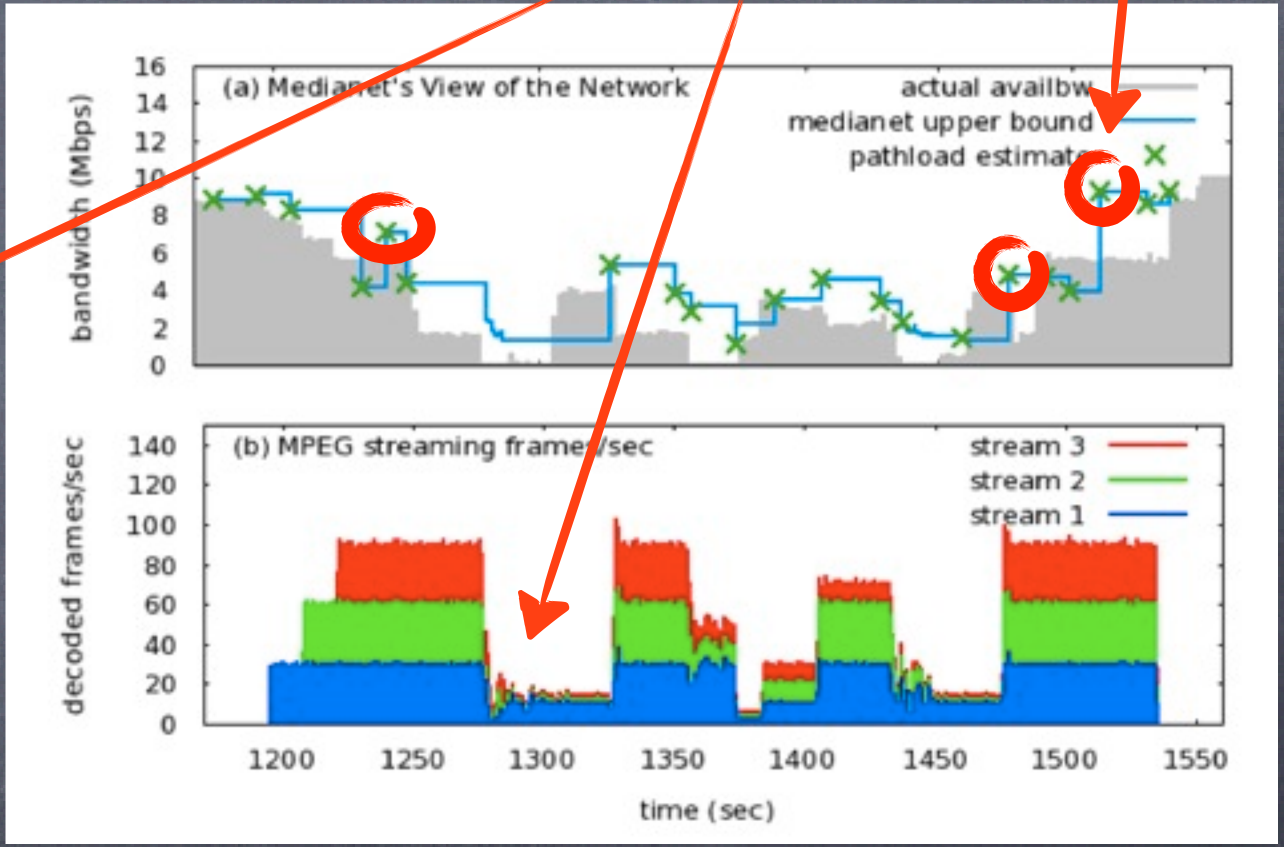
available bandwidth known to global scheduler

cross traffic fills in available bandwidth

regular estimates from pFAST



Original MediaNet



MediaNet with pFAST over MGRP

mediaNet more closely follows the actual available bandwidth

Measurement Overlay

MediaNet Results

relative % improvement of
using overlay!

without MGRP: pSLOW > pFAST

experiment	runs	sec	Mbps	inc. over mgrpOFF	inc. over original	fps	inc. over mgrpOFF	inc. over original
mgrpOFF.pOFF	14	337	1.84			30.11		
mgrpOFF.pSLOW	22	336	1.96		6.29%	39.58		31.44%
mgrp10.pSLOW	32	336	2.05	**4.40%**	11.21%	43.42	**9.69%**	44.19%
mgrpOFF.pFAST	10	335	1.86		0.94%	39.10		29.87%
mgrp10.pFAST	22	336	2.28	**22.52%**	23.86%	52.08	**33.19%**	72.96%

improvement of
using active probes

increase in streaming rates with Measurement Overlay

Advantages

- flexibility and accuracy of active probing + low overhead of passive probing
- minimal changes to existing code of probe tools
- no changes to applications

Disadvantages

- no bandwidth saving when no user data.
- increases complexity of transport.
- kernel-specific: harder to deploy.

References

- Papageorge and McCann, Passive aggressive measurement with MGRP. ACM SIGCOMM Computer (2009)
- Papageorgiou. The Measurement Manager: Modular End-to-End Measurement Services. (2007)
- Papageorgiou, Merging Network Measurement with Data Transport. Passive and Active Network Measurement (2005)
- De Cicco and Mascolo, Skype video responsiveness to bandwidth variations for Digital Audio and Video (2008)
- Chu et al. A case for end system multicast. IEEE Journal on Selected Areas in Communications (2002)
- Vishal and Misra, SOS: Secure overlay services. In Proceedings of ACM SIGCOMM (2002)