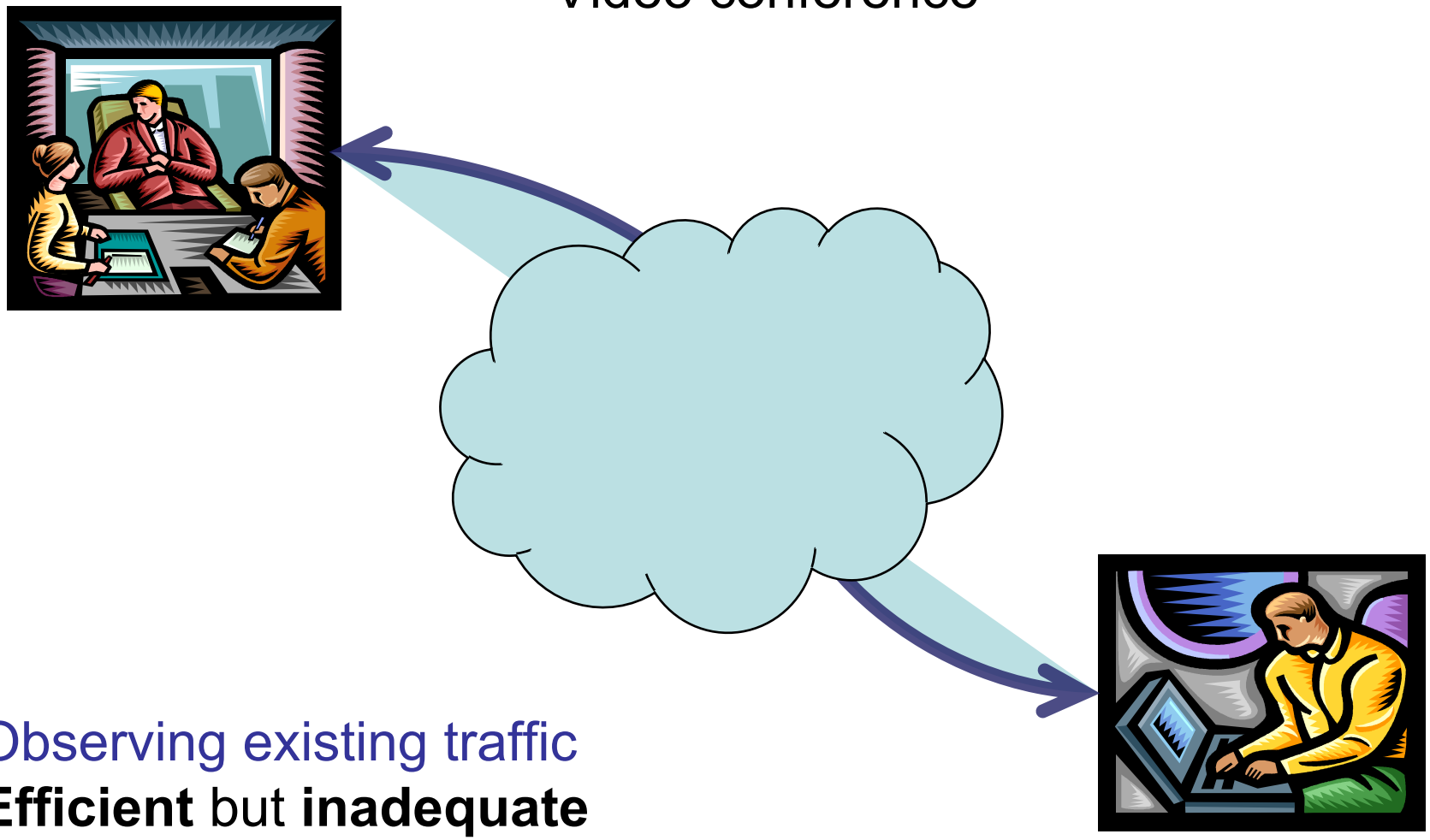
Passive Aggressive Measurement with MGRP

Pavlos Papageorge[1,2], Justin McCann[2], Michael Hicks[2]

Google[1]
University of Maryland, College Park[2]

Choice 1: Passive Measurement

Video conference



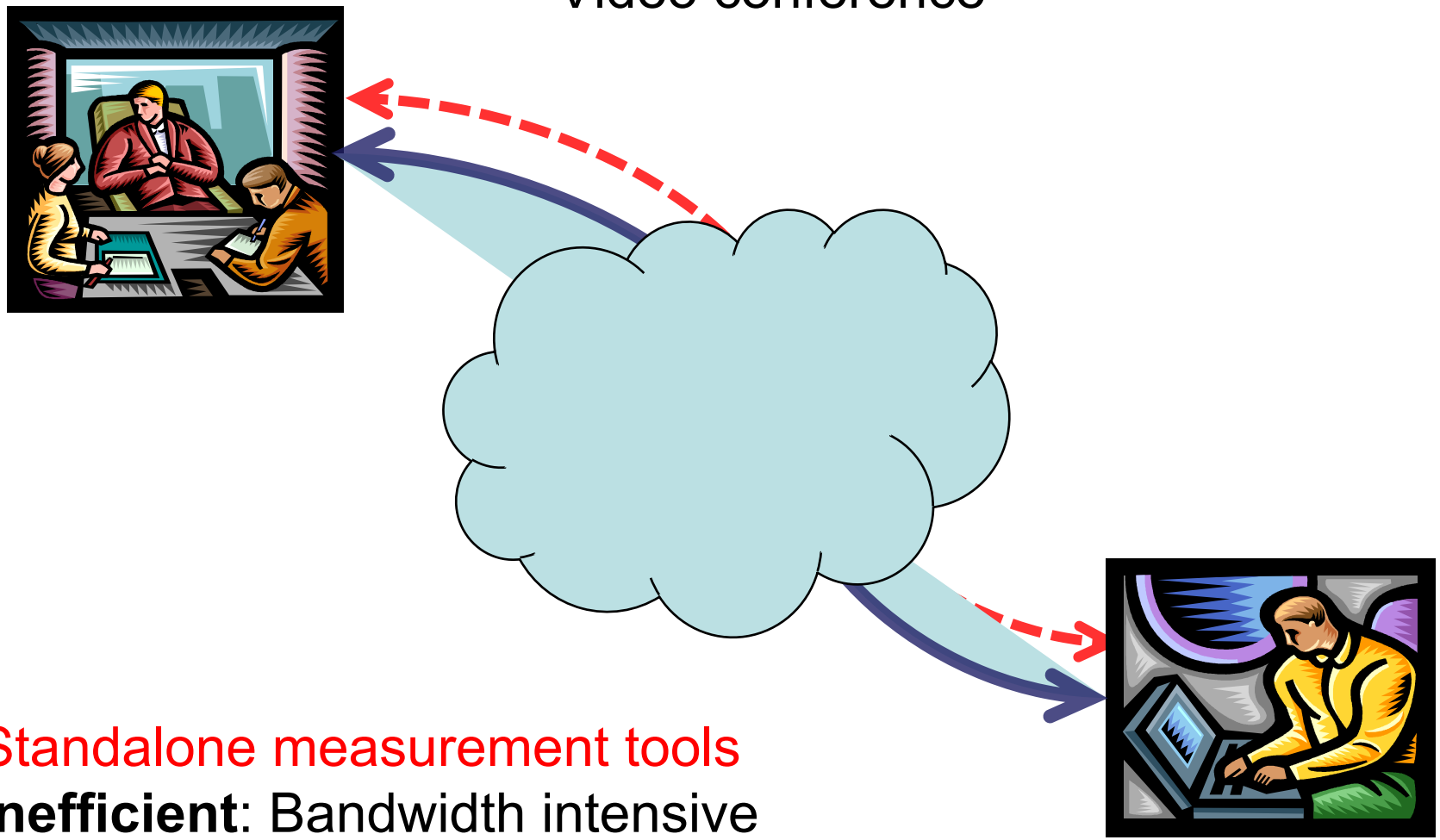
Observing existing traffic

Efficient but **inadequate**

Cannot detect when network conditions improve

Choice 2: Active Measurement

Video conference



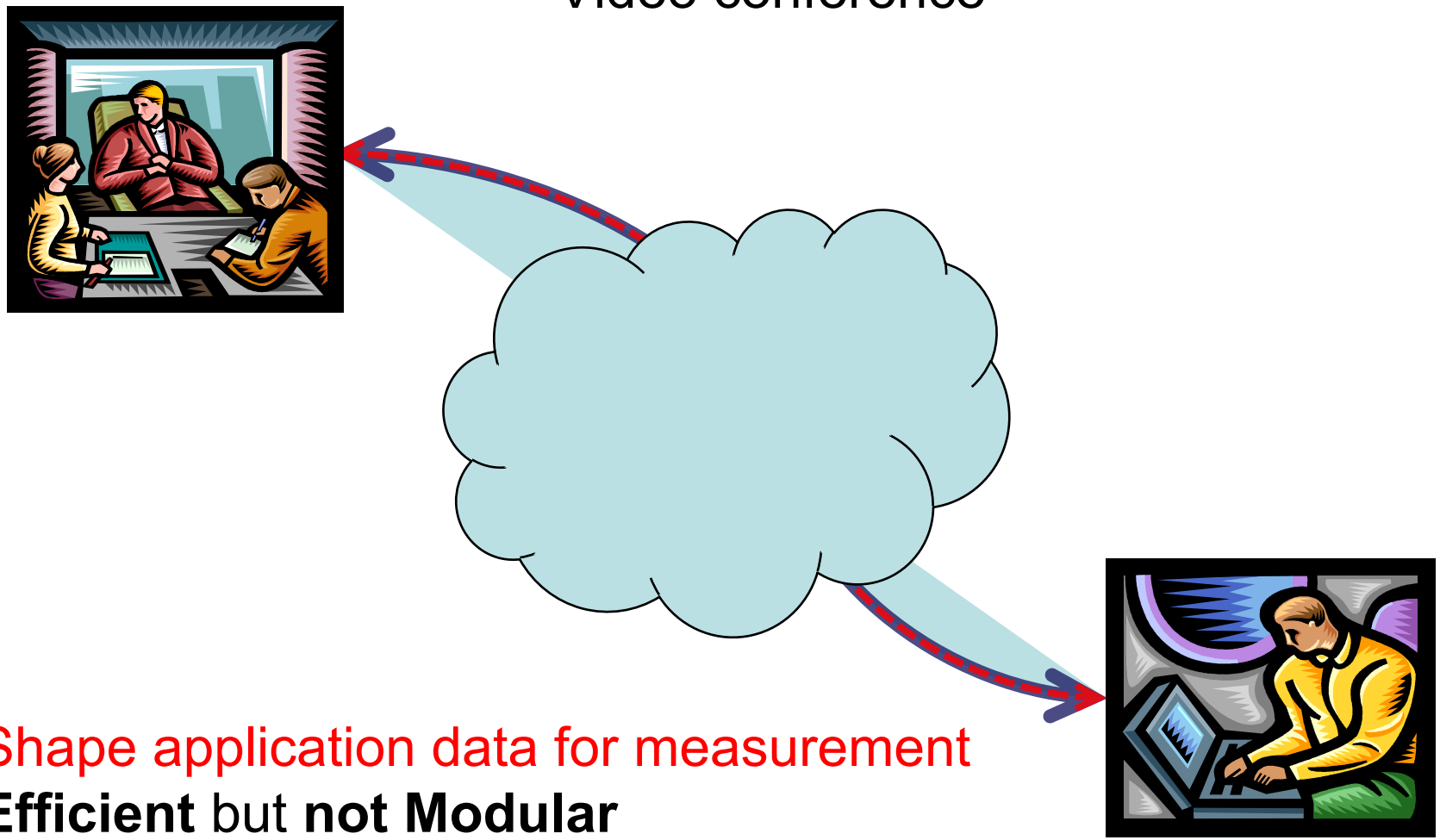
Standalone measurement tools

Inefficient: Bandwidth intensive

Intrusive: Probes can interfere with application data

Choice 3: Custom Active Measurement

Video conference

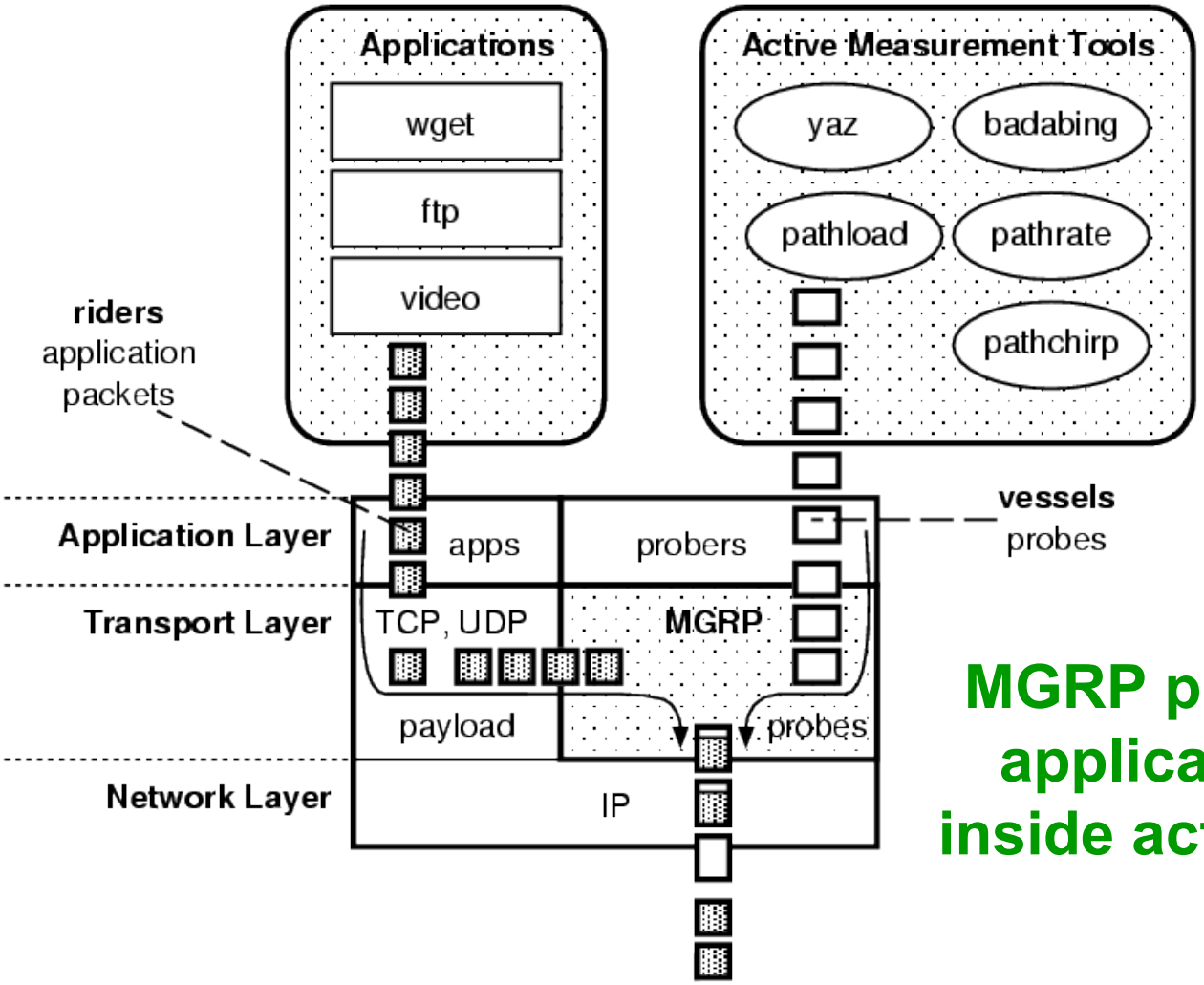


Shape application data for measurement

Efficient but **not Modular**

Not Reusable: Cannot interchange algorithms

MGRP
The Measurement Manager Protocol



MGRP piggybacks application data inside active probes

MGRP Properties

Enables aggressive probing with passive-like overhead

- End-to-end measurement architecture
  - Schedules probes for transmission
  - Piggybacks application data on probes
- Transparent to applications
- Independent of measurement algorithms
- Easy to adapt existing measurement tools
- Can piggyback data across applications
- μsec precision for probe gap generation

Outline

Motivation

Why do we need MGRP?

MGRP

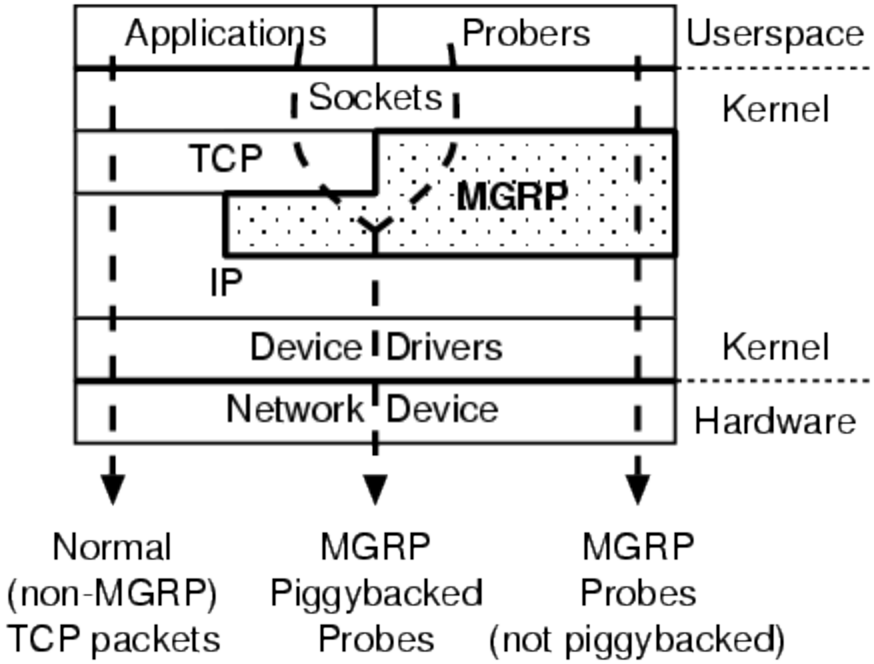
Architecture
Implementation
Step-by-Step Examples

Micro-Benchmarks

Piggybacking is feasible and improves network performance

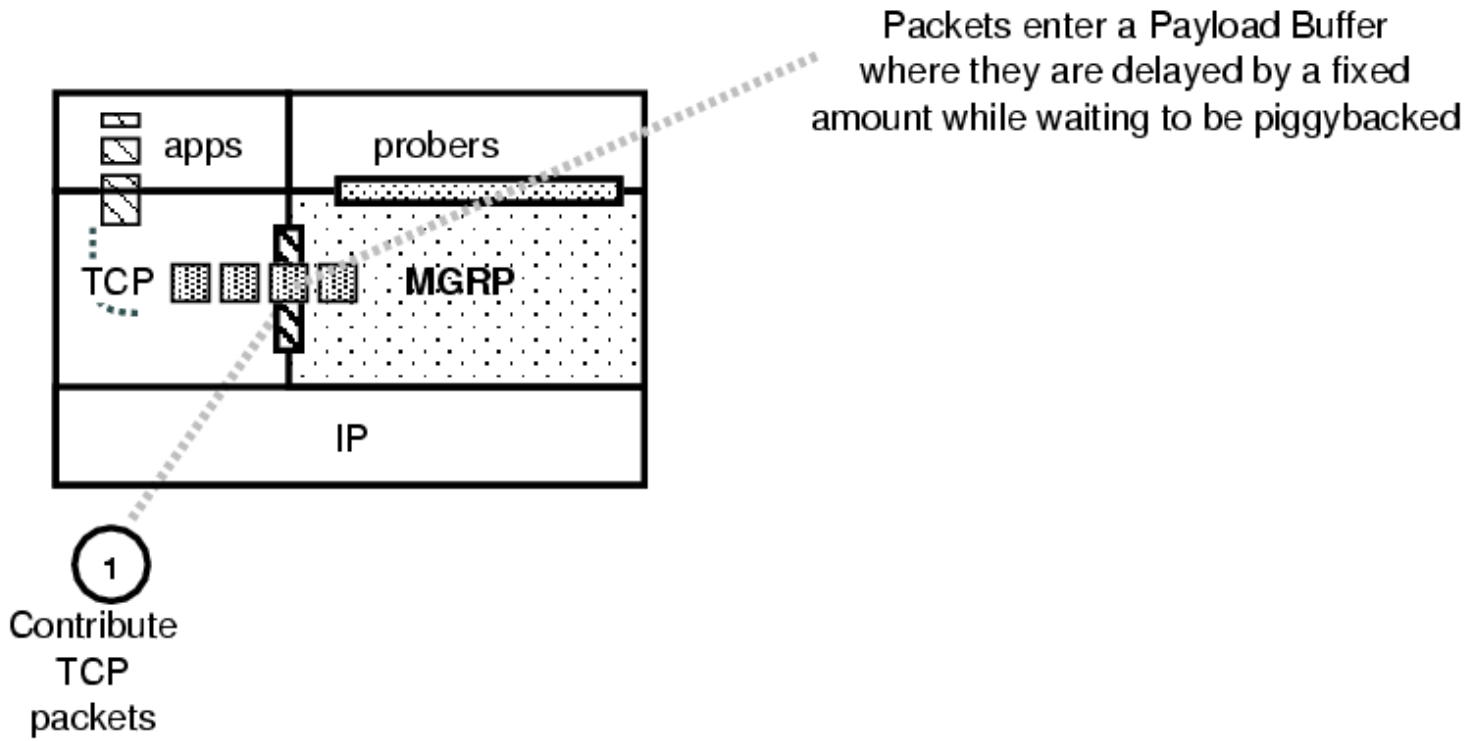
Case Study: MediaNet

MGRP in the Network Stack



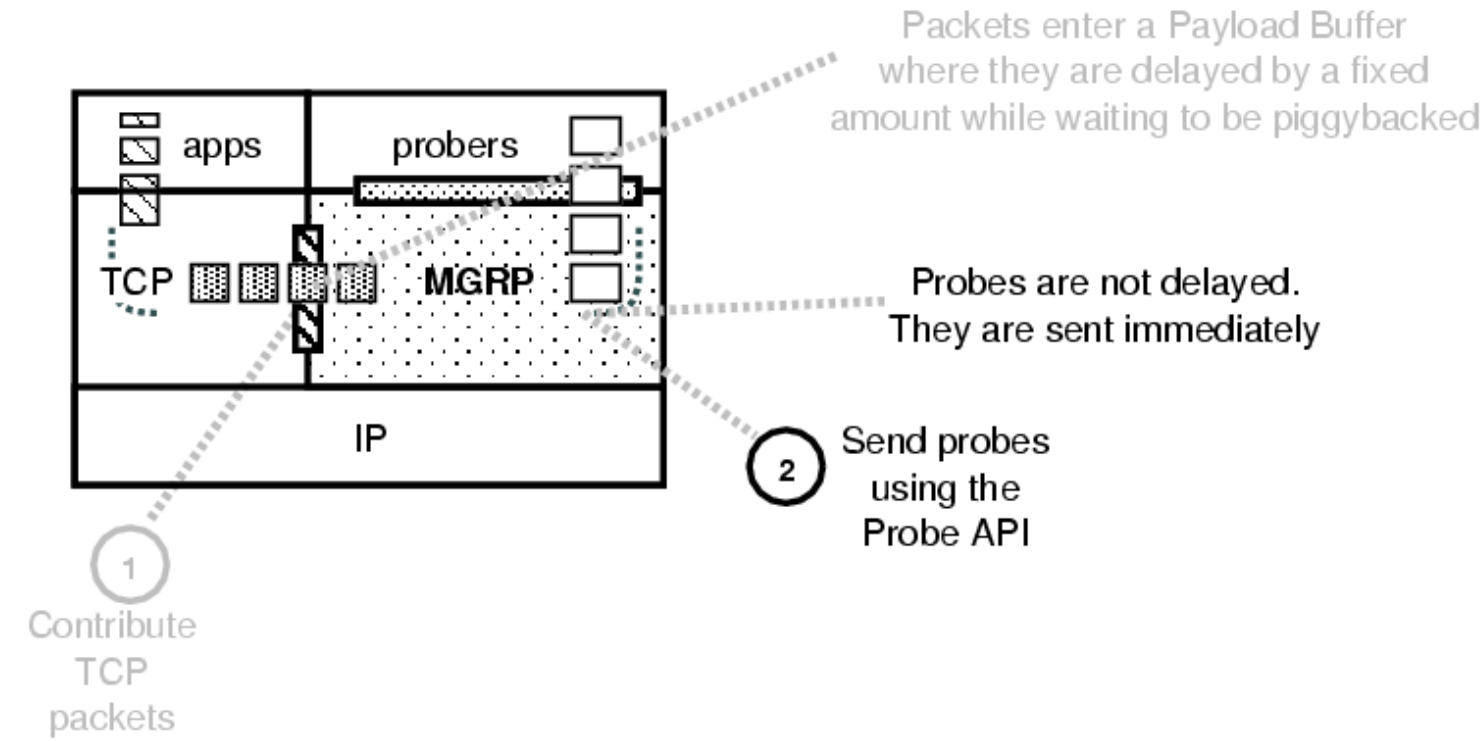
- Layer 4 transport protocol
- Implemented in the Linux kernel

MGRP: Step by Step Example



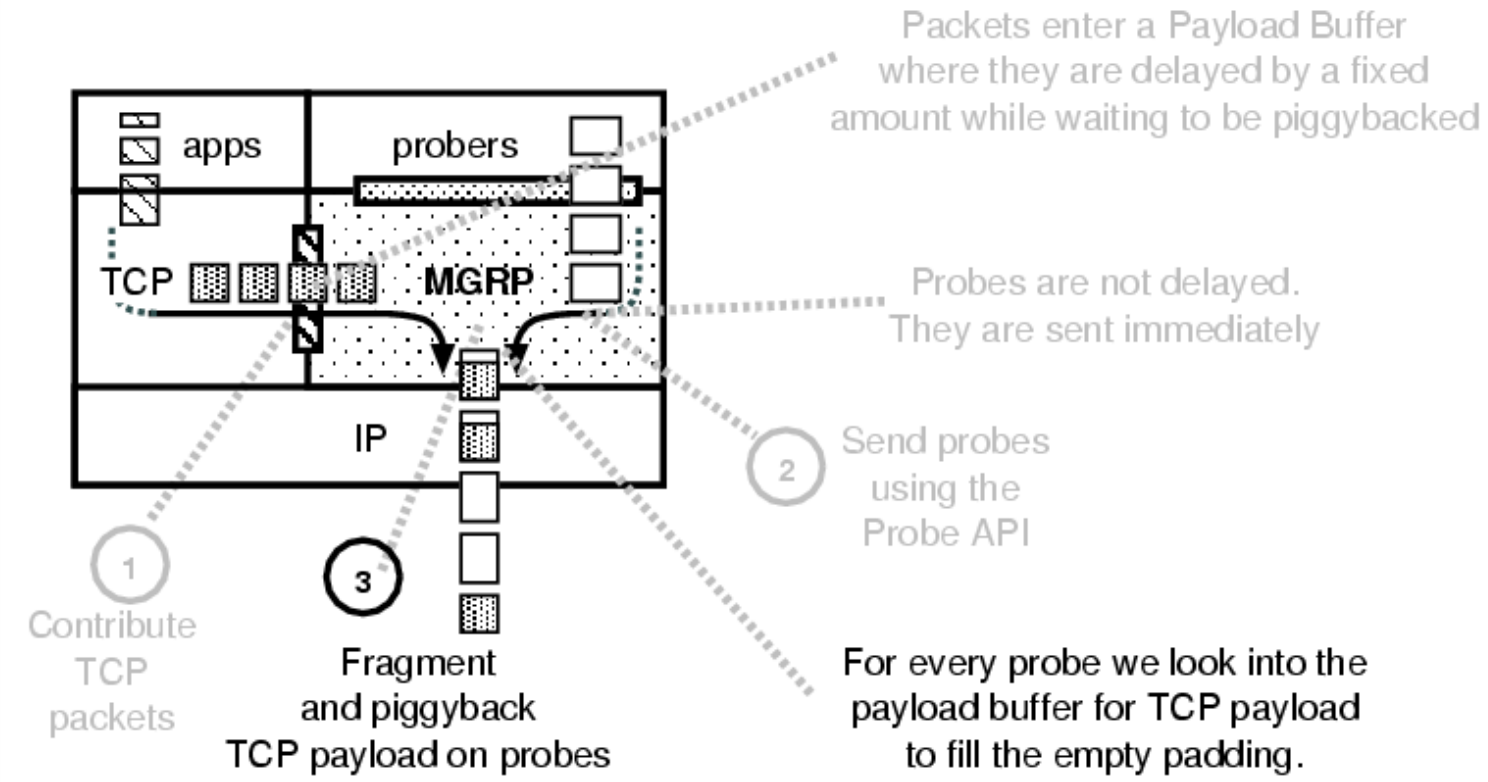
MGRP Sender piggybacks payload on probes

MGRP: Step by Step Example



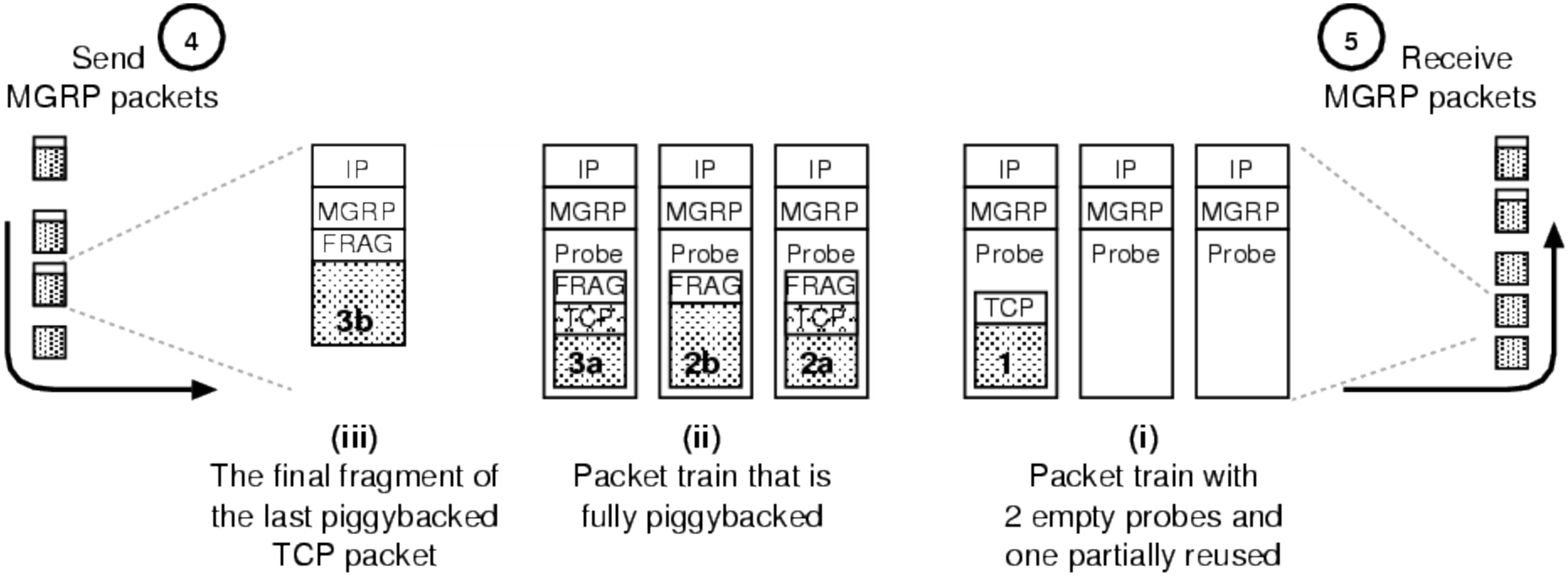
MGRP Sender piggybacks payload on probes

MGRP: Step by Step Example



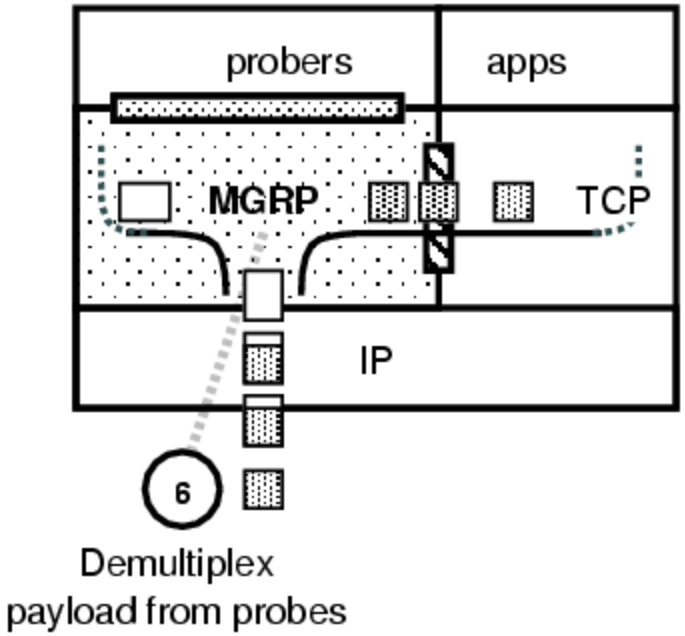
MGRP Sender piggybacks payload on probes

MGRP: Step by Step Example



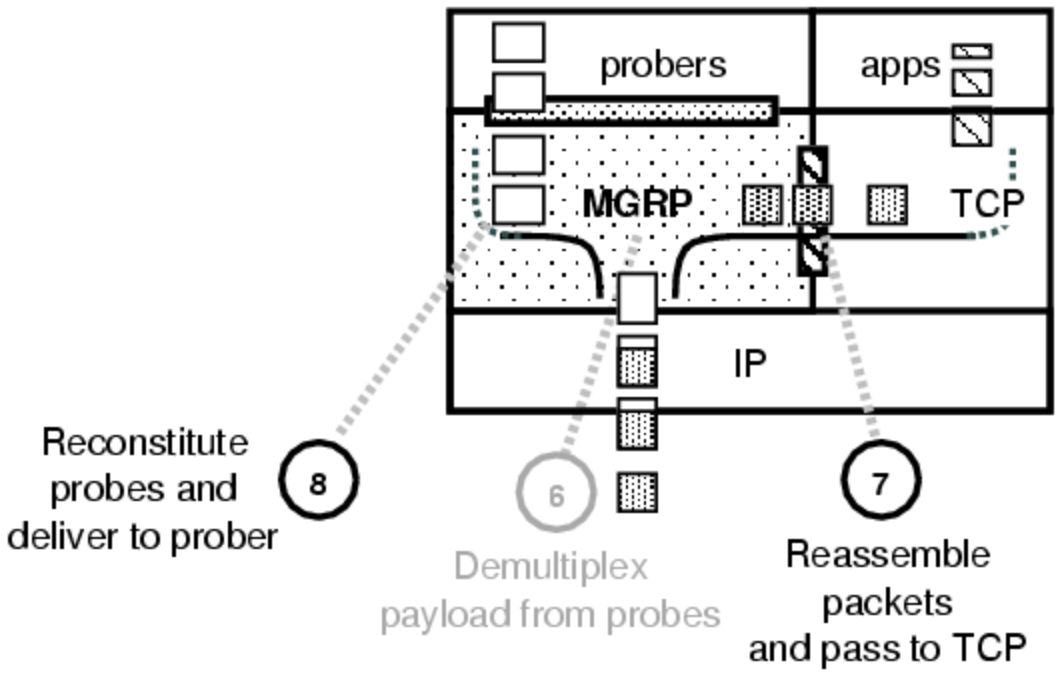
MGRP packets traverse the network

MGRP: Step by Step Example



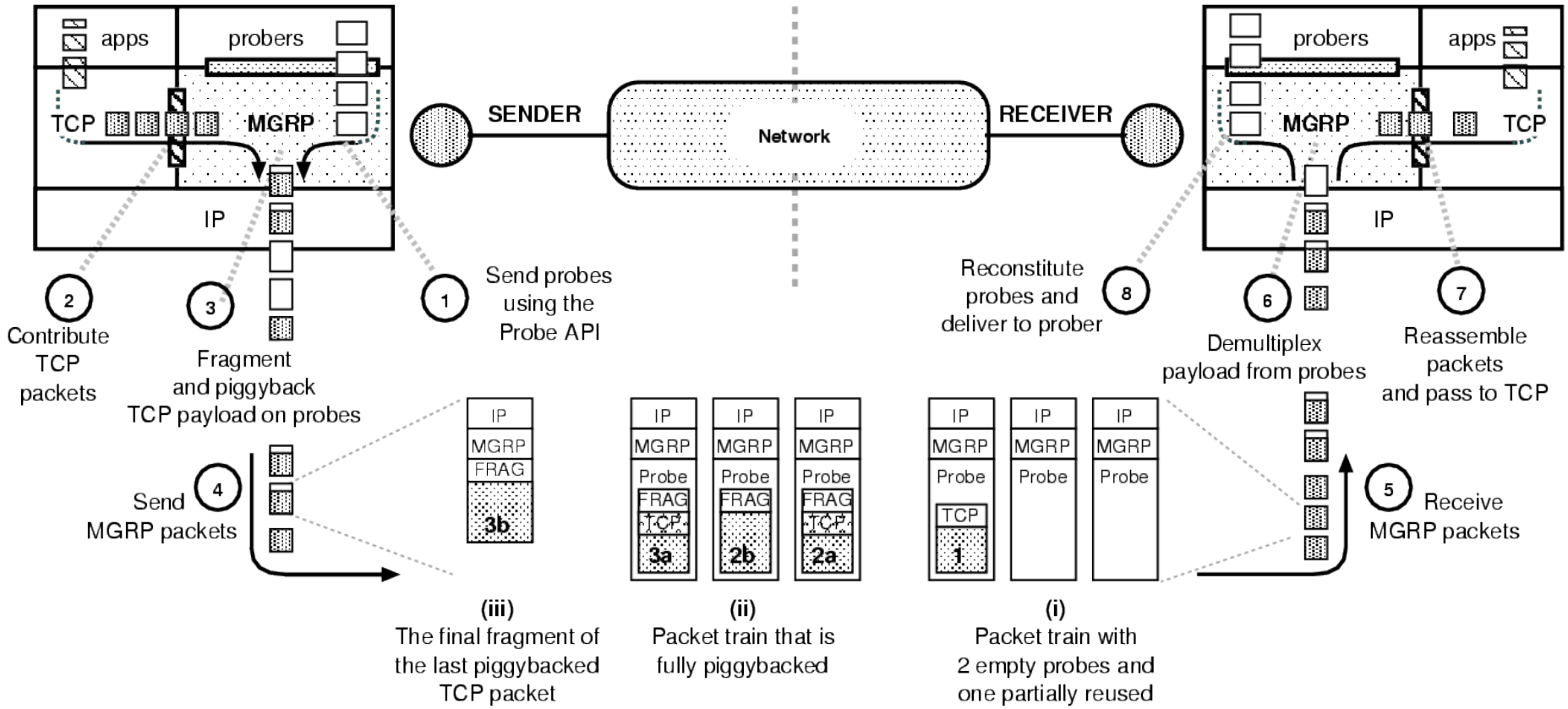
MGRP Receiver reconstitutes probes and payload

MGRP: Step by Step Example



MGRP Receiver reconstitutes probes and payload

MGRP: Step by Step Example



Outline

Motivation

Why do we need MGRP?

MGRP

Architecture
Implementation
Step-by-Step Examples

Micro-Benchmarks

Piggybacking is feasible and improves network performance

Case Study: MediaNet

Available Bandwidth Tool

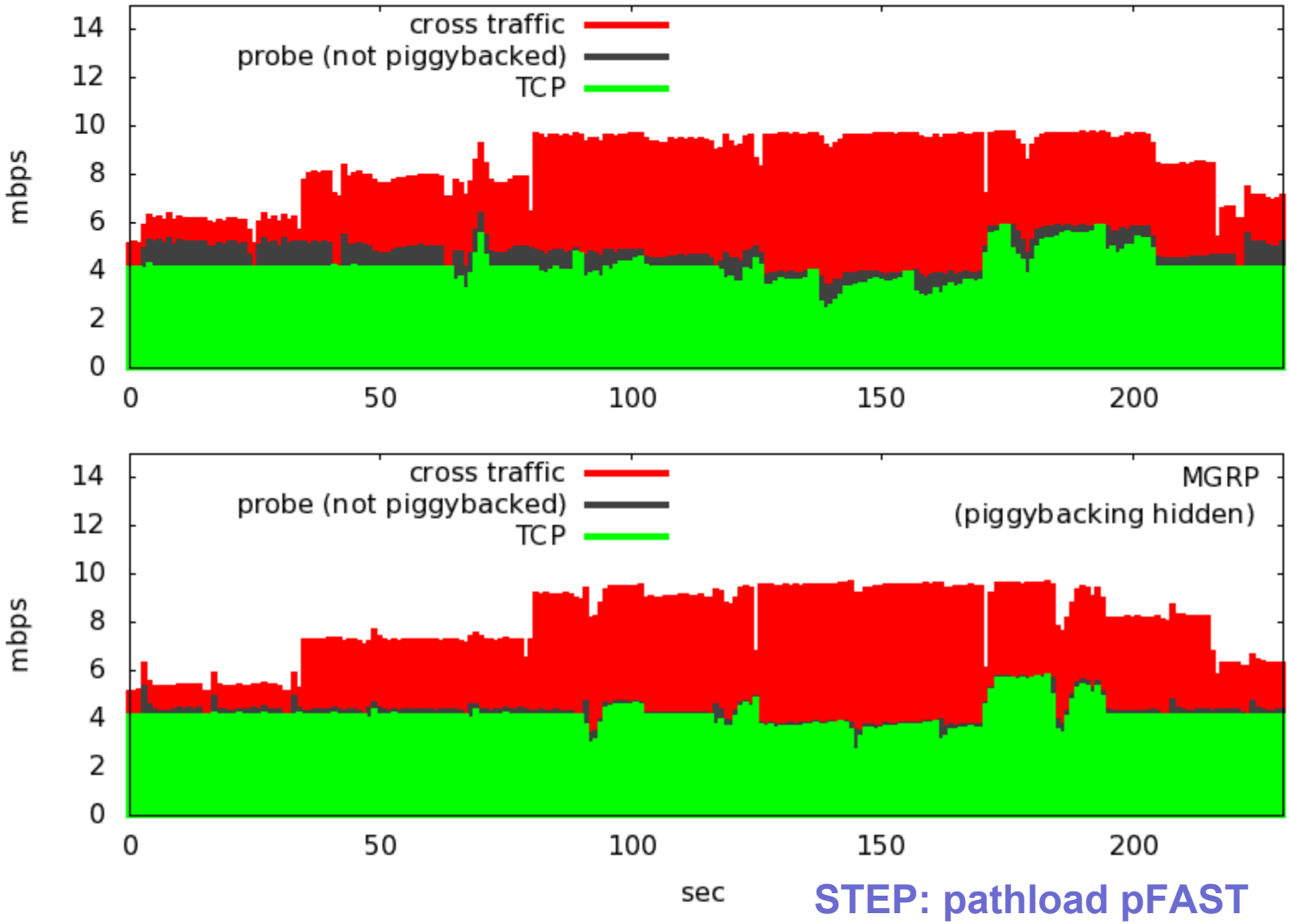
- **Pathload**: An active measurement tool
  - Measures end-to-end available bandwidth
  - By Jain & Dovrolis at Georgia Tech

- Good candidate for our evaluation
  - Available bandwidth is a very useful network property
  - Quite accurate (even for GigE speeds, PAM05)
  - Non-trivial overhead (we can test probe reuse)

Bandwidth Timeseries with Pathload

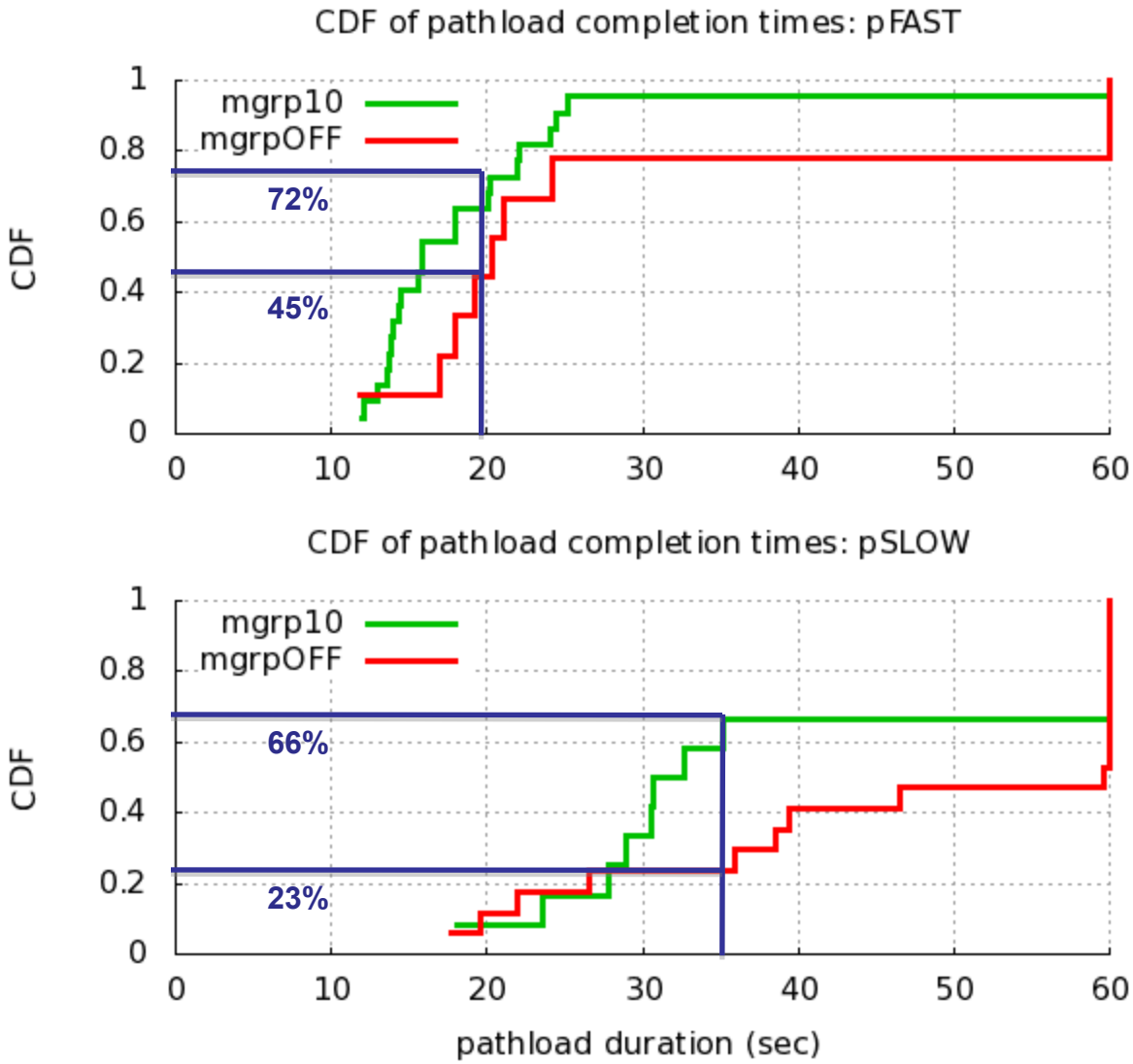
STEP: pathload pFAST

Effective Probe Overhead is Minimal



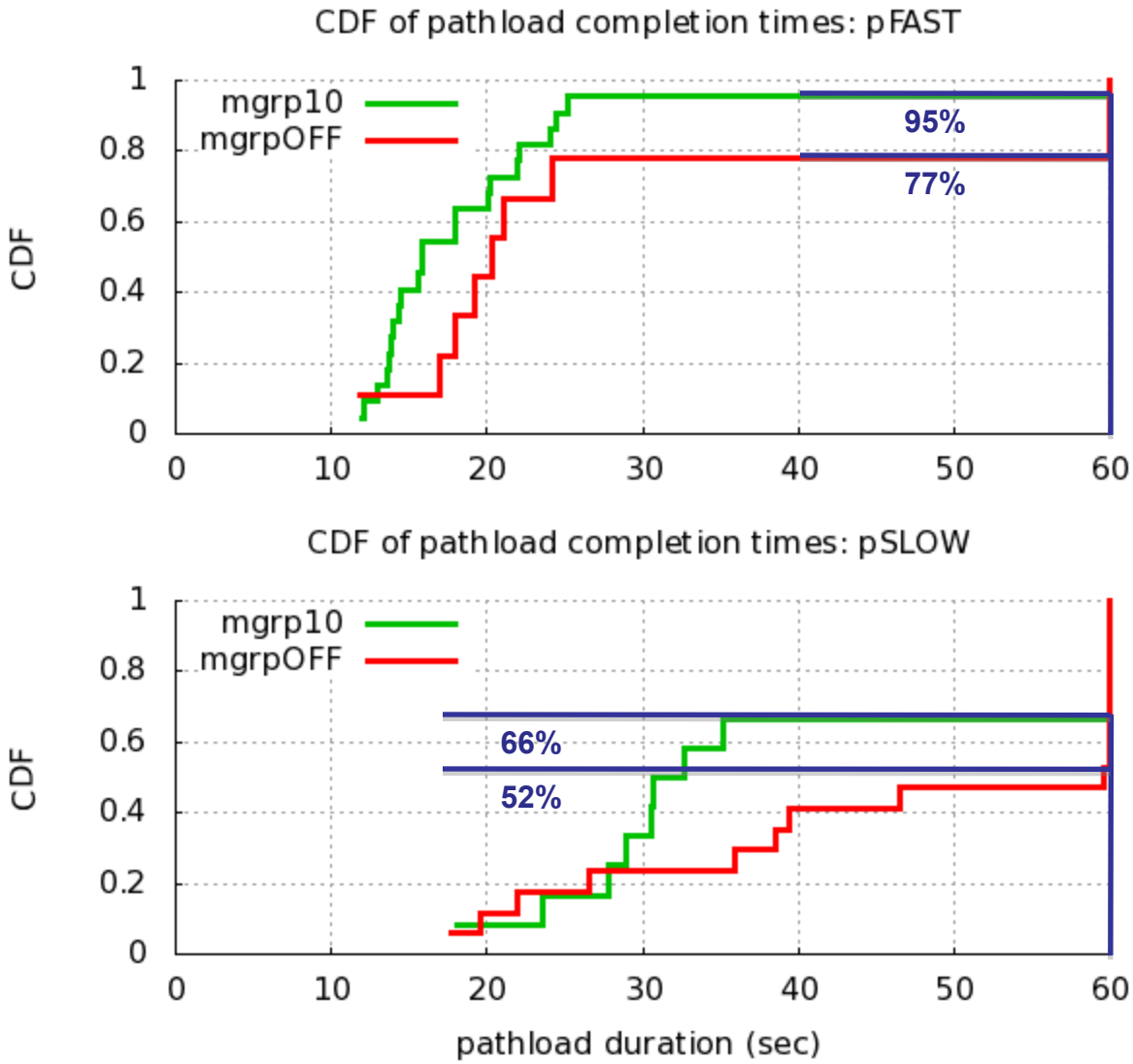
STEP: pathload pFAST

Pathload Completes Faster



STEP: pathload

Pathload Completes More Often



STEP: pathload

Benefits of MGRP

- Saves bandwidth
  - Reduces measurement overhead
  - Fewer probes compete with application data

- Allows measurement tools to:
  - Send more probes
  - Send probes continuously
  - Complete faster and be more accurate

Lessons Learned

- Measurement tools need to be adjusted
  - Must account for piggybacked traffic
- Blind piggybacking can be harmful
  - Pigybacked packets share fate of probes
  - Some probes have high loss risk
- Long MGRP data buffers may affect TCP
  - Need to keep latency small fraction of RTT

Outline

Motivation

Why do we need MGRP?

MGRP

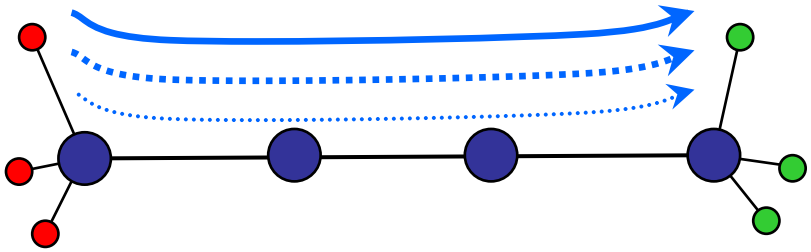
Architecture
Implementation
Step-by-Step Examples

Micro-Benchmarks

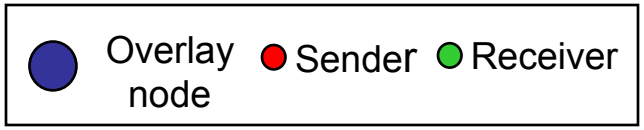
Piggybacking is feasible and improves network performance

Case Study: MediaNet

Case Study: MediaNet Overlay



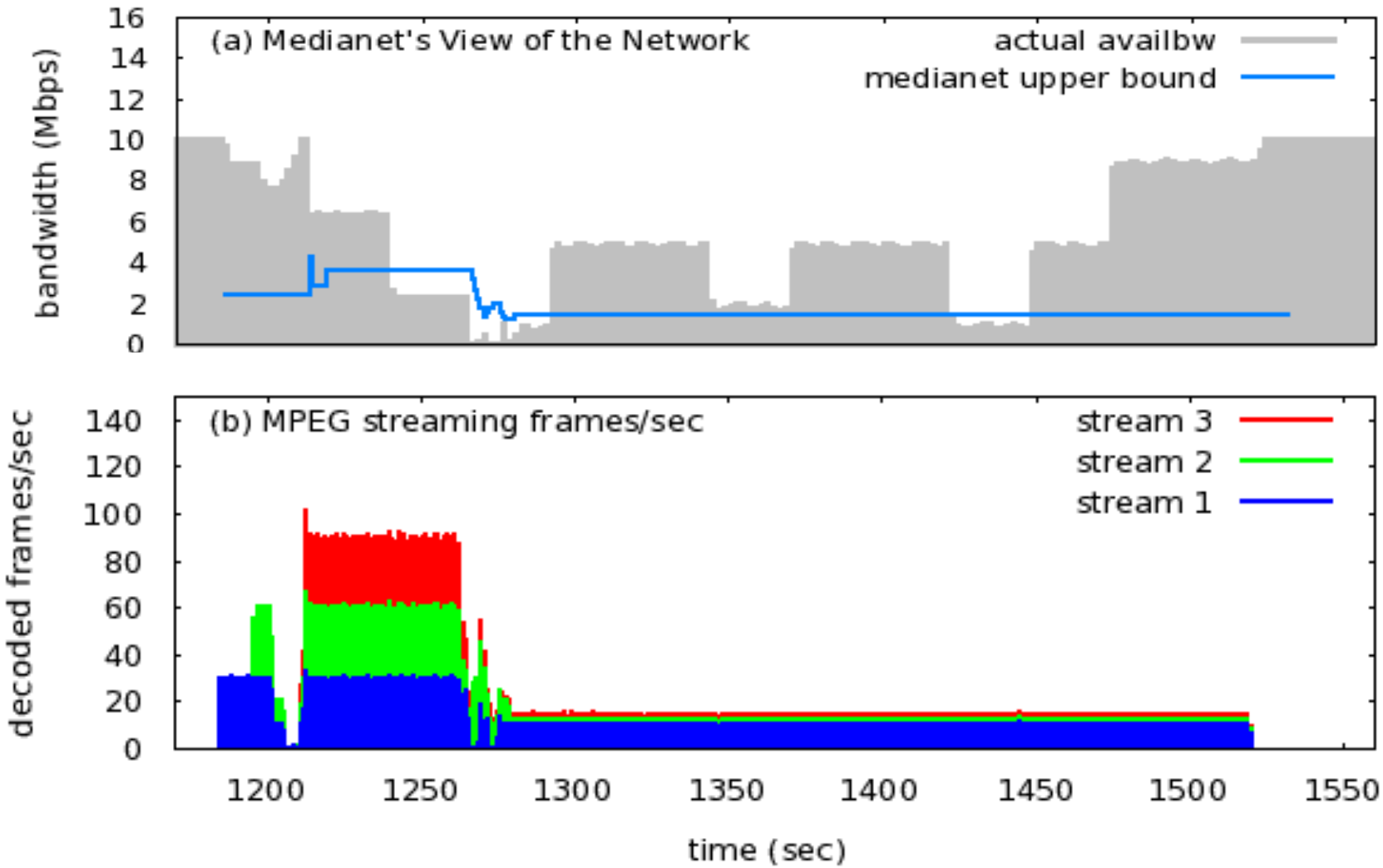
(a) MediaNet can modify the streaming rate



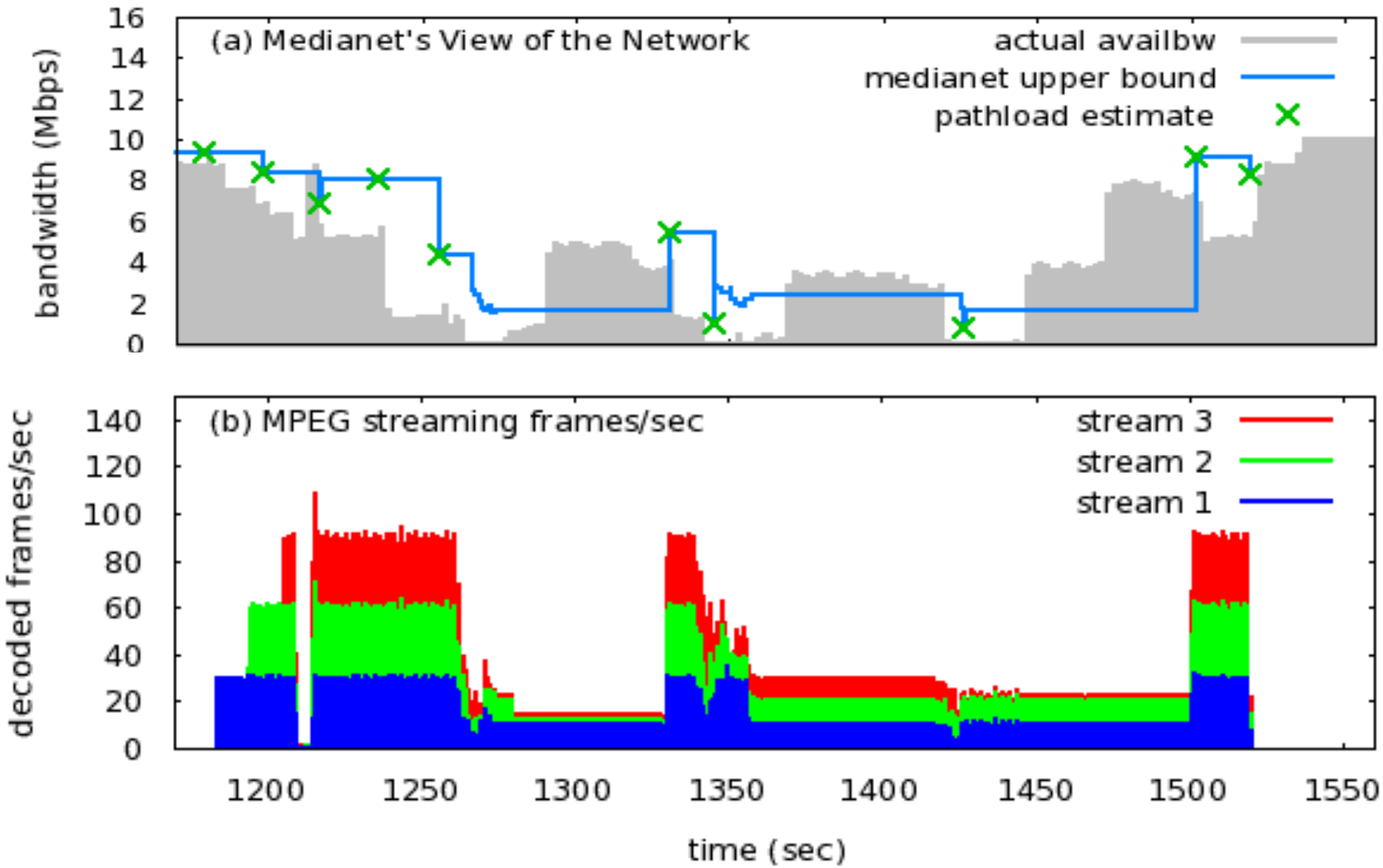
- Streams MPEG video at different rates
- **Overlay nodes** report if they can send at desired rate
- **Pathload** continuously monitors the paths

Frame Type	Frame Size (bytes)	Frequency (frames/sec)	High Rate (Kbps)	Medium Rate (Kbps)	Low Rate (Kbps)
I	13500	2	1200	700	200
P	7625	8			Dropped
B	2850	20		Dropped	Dropped

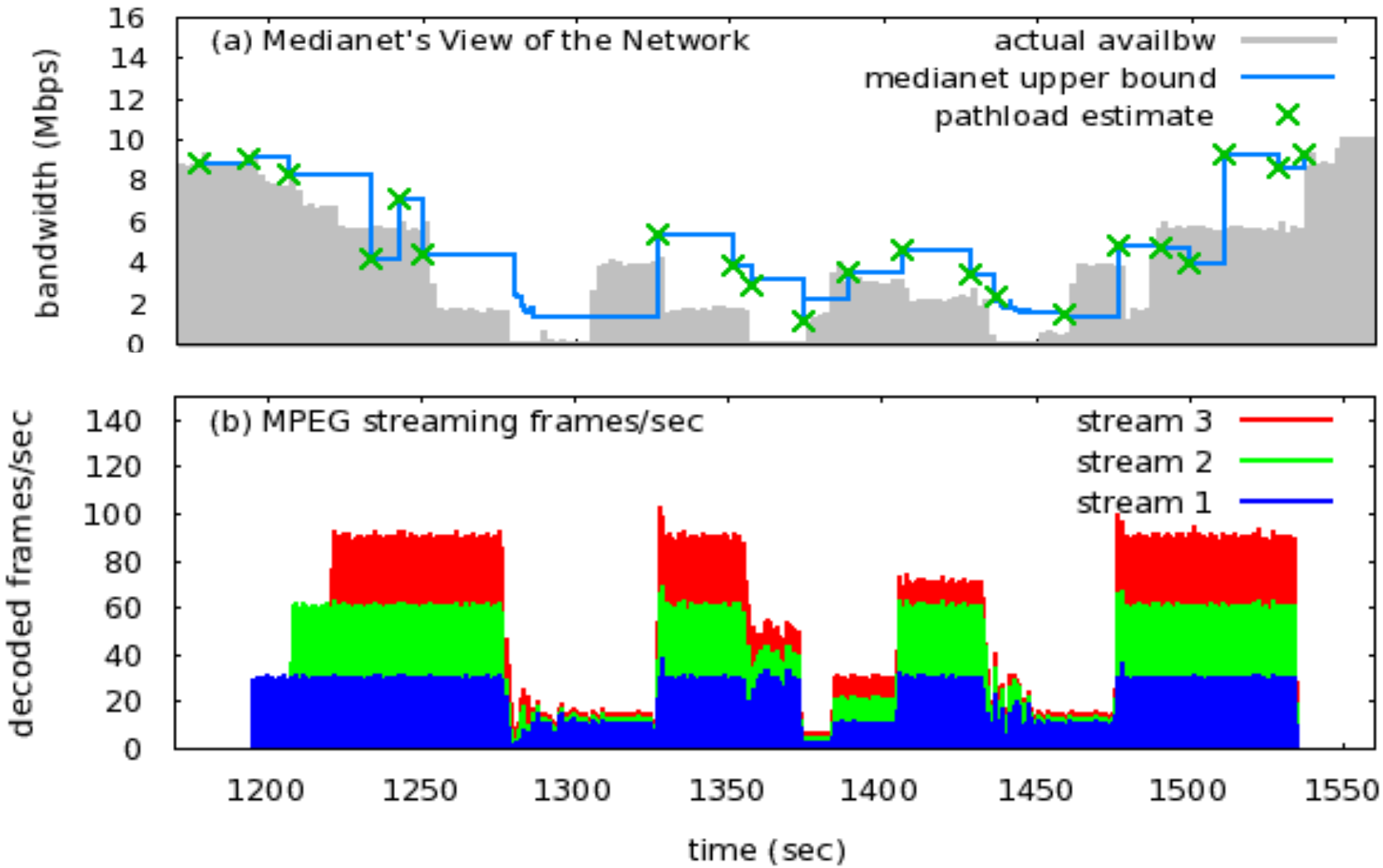
Without probes MediaNet cannot react



But probes interfere without MGRP



MGRP improves the stream quality



MGRP Improves MediaNet

The aggregate MPEG streaming rate is higher

Experiment		Runs	AverageRun Duration (sec)	Aggregate Streaming Rate (Mbps)	Improvement over non-MGRP
Original MediaNet (mgrpOFF/pOFF)		14	337	1.84	
Pathload pSLOW	mgrpOFF	22	336	1.96	
	mgrp10	32	336	2.05	**4.40%**
Pathload pFAST	mgrpOFF	10	335	1.86	
	mgrp10	22	336	2.28	**22.52%**

Related to the quality of the playback

MGRP Improves MediaNet

The number of decoded MPEG frames increases

Experiment		Runs	AverageRun Duration (sec)	Aggregate Frames per Second	Improvement over non-MGRP
Original MediaNet (mgrpOFF/pOFF)		14	337	30.11	
Pathload pSLOW	mgrpOFF	22	336	39.58	
	mgrp10	32	336	43.42	**9.69%**
Pathload pFAST	mgrpOFF	10	335	39.10	
	mgrp10	22	336	52.08	**33.19%**

Directly affects the quality of the playback

How MGRP stands out

MGRP is a new protocol that piggybacks application data inside probes.

Piggybacking reduces bandwidth wasted by probes and enables measurement tools to be more aggressive, faster and more accurate.

Any measurement algorithm can now be written as if active, but implemented as passive.

MGRP is generic and is transparent to applications

Questions?

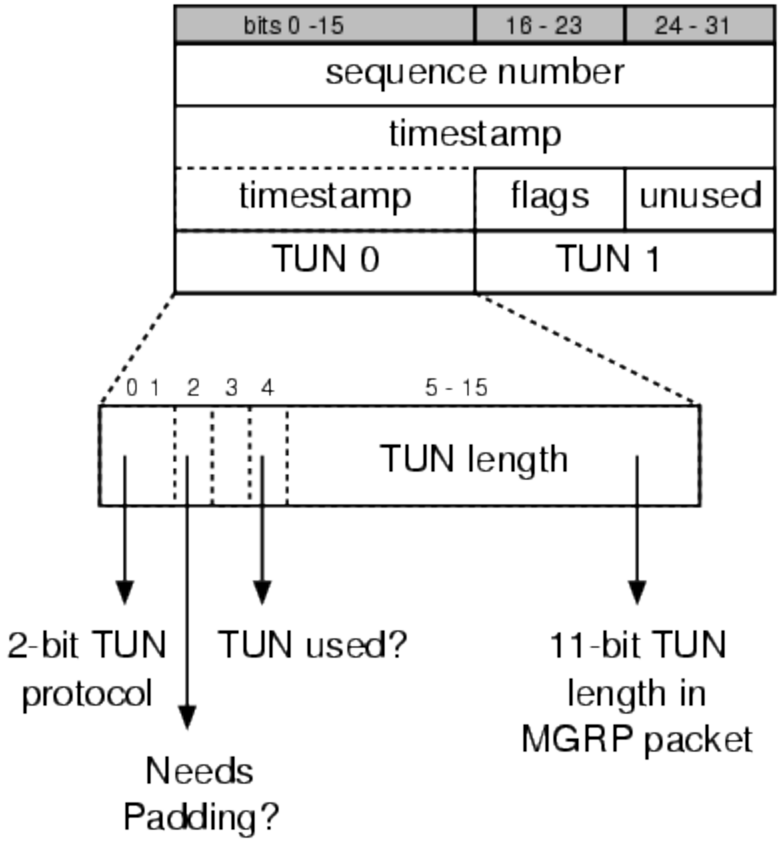
Additional Slides

MGRP: Next Steps

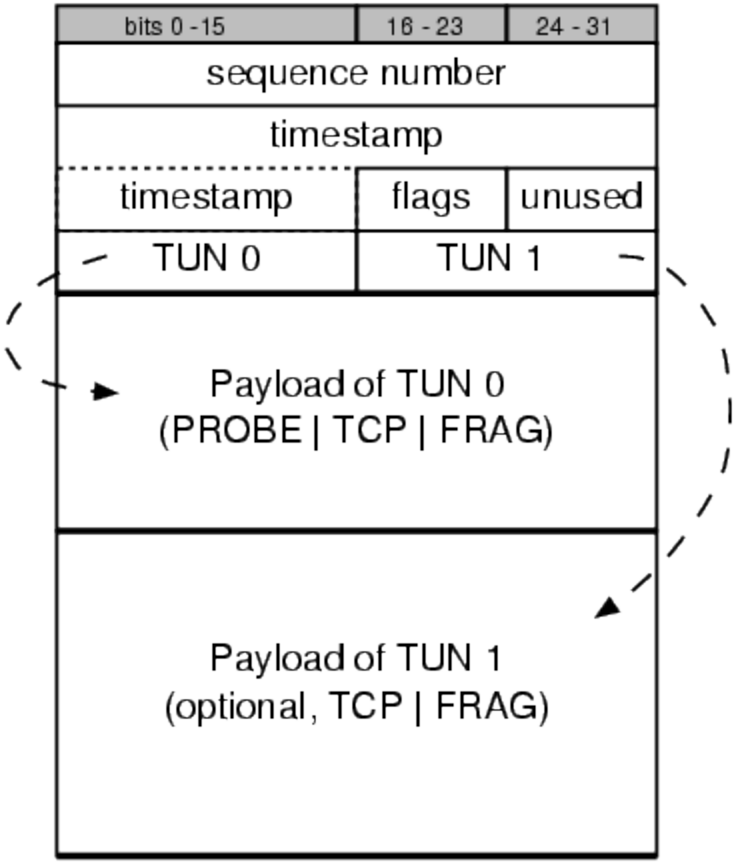
- Add support for:
  - ICMP packets
  - TTL limited packets
- Automatically set piggybacking ratio
- Enable one-way probing with remote timestamp collection

MGRP Packet Format

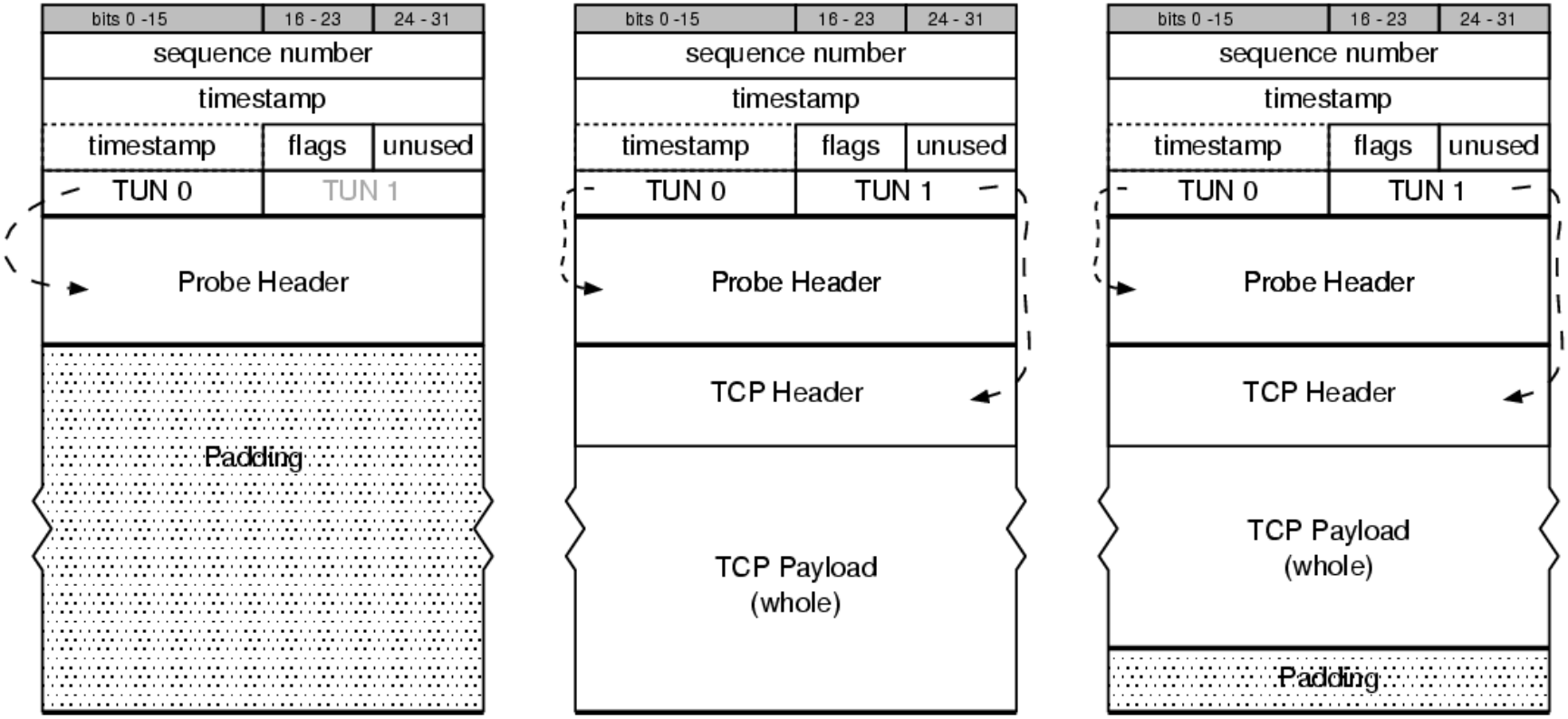
MGRP Header



MGRP Packet



MGRP Piggybacking



(a) MGRP probe with no piggybacking

(b) MGRP probe fully piggybacked

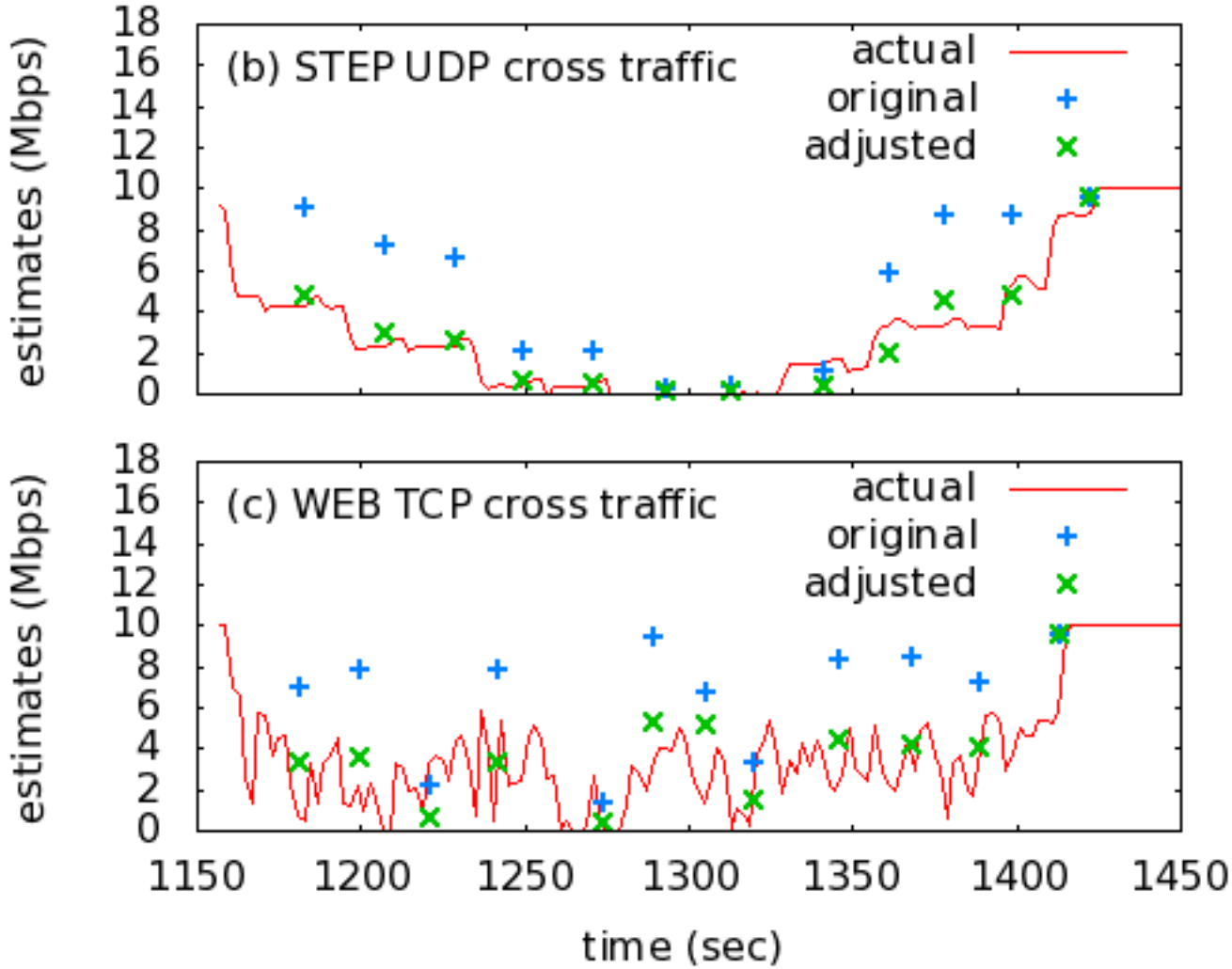
(c) MGRP probe partially piggybacked

Characteristics of Active Probes

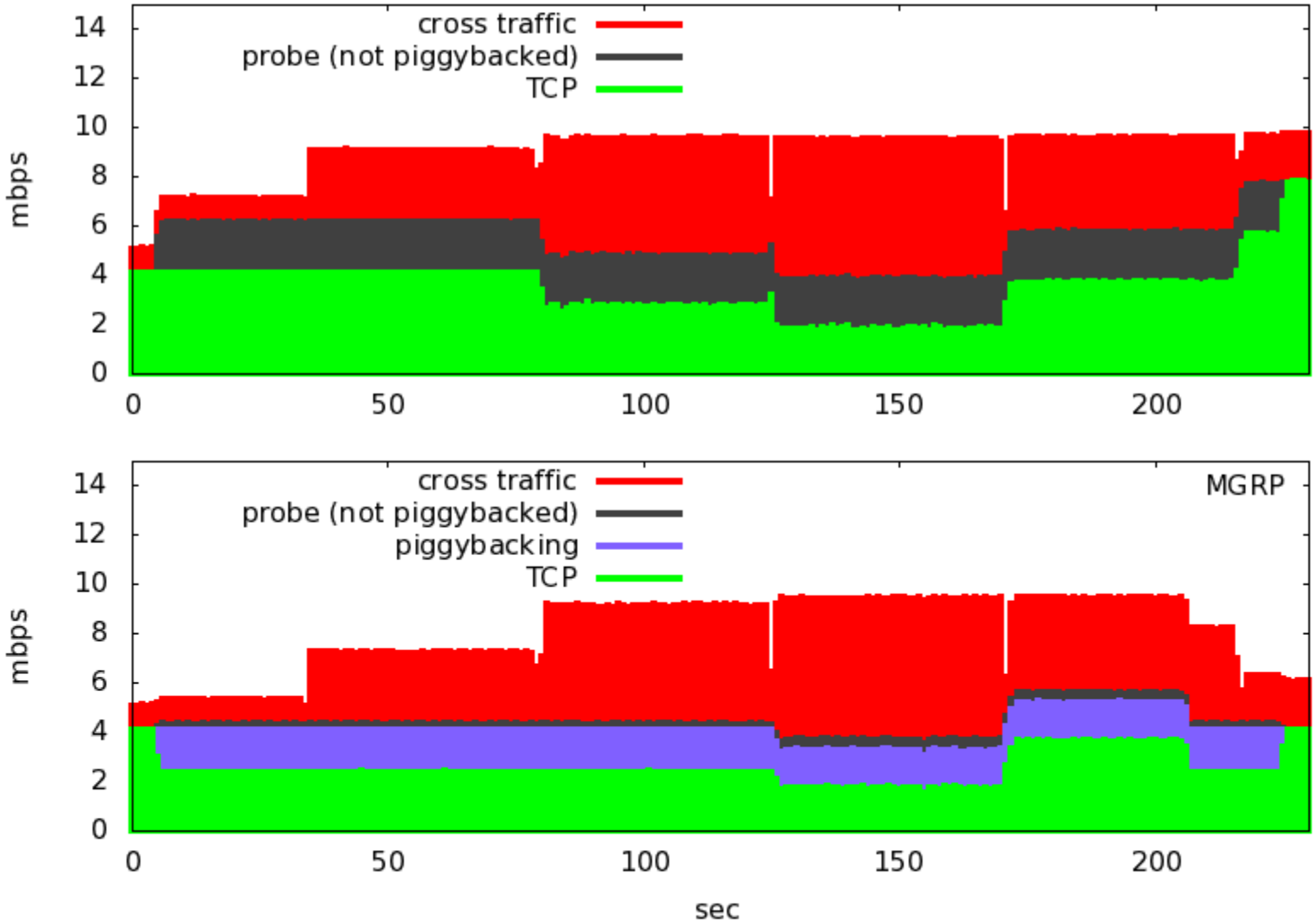
- Have varying sizes
- Need precise inter-packet gaps
- Are largely empty padding
- Usually sent in groups
- More probes: better/faster results

- Can probing be aggressive without the overhead?

Piggybacking requires that we adjust Pathload

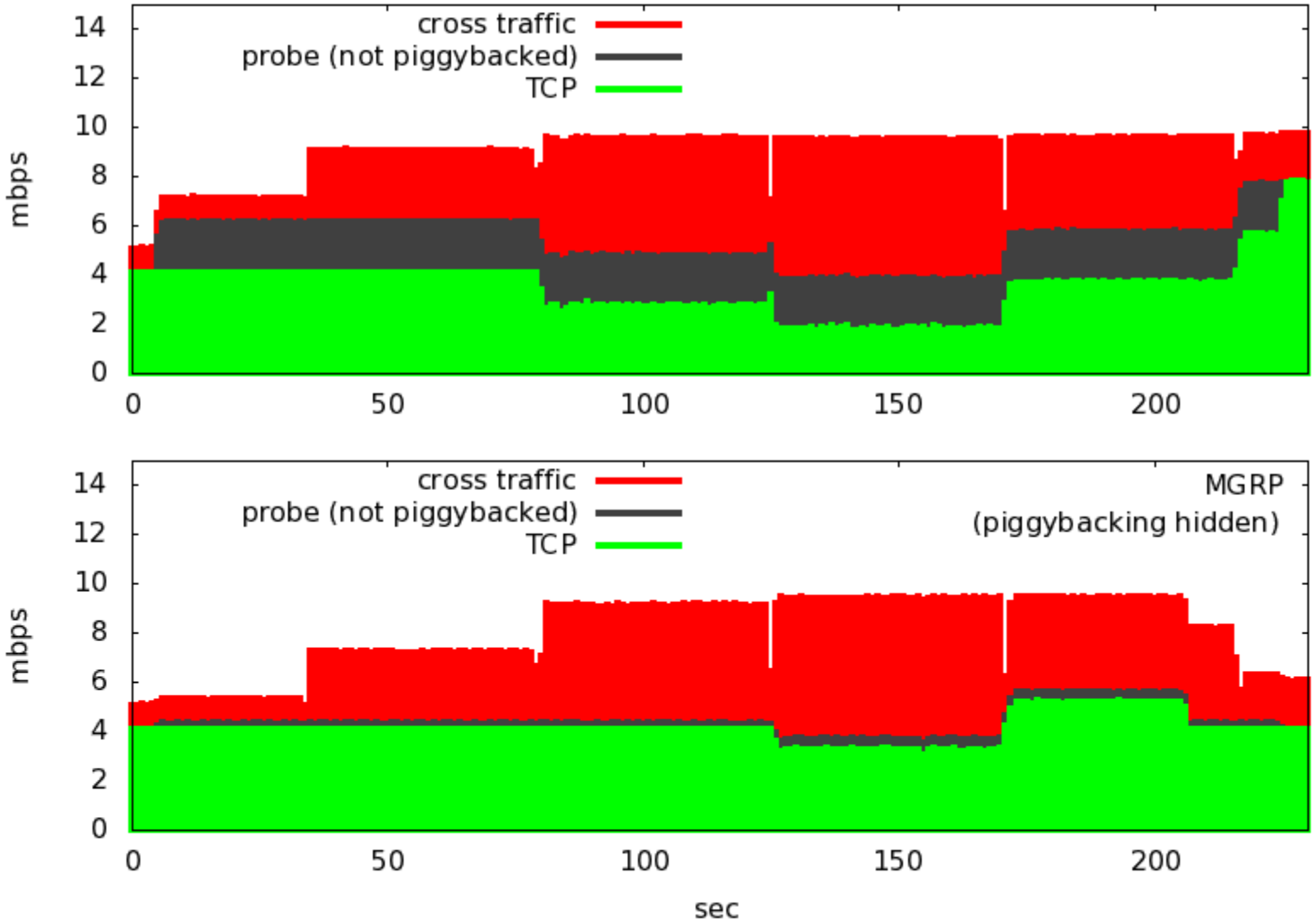


Piggybacking reduces the probing overhead



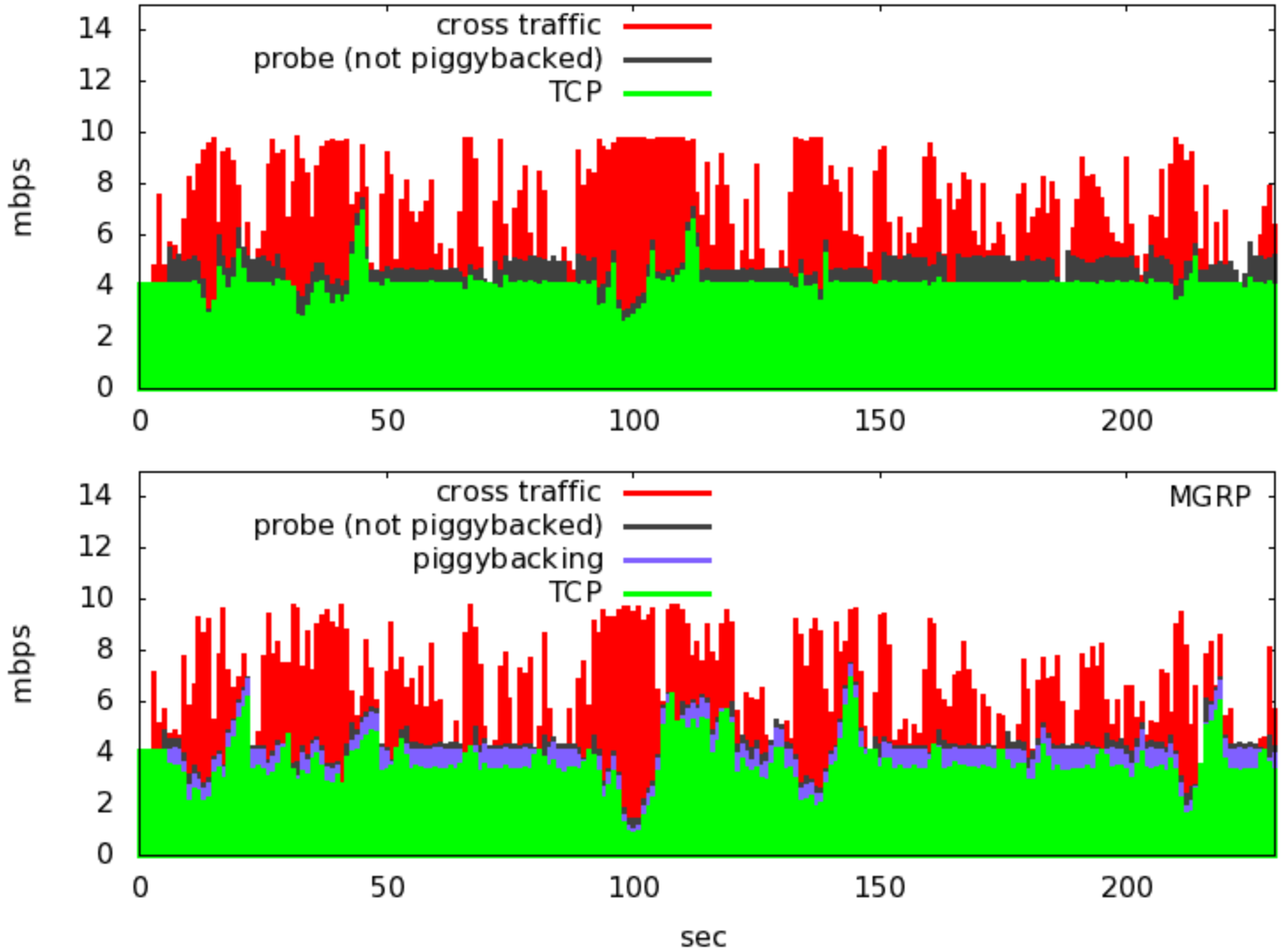
STEP: probe train pk2

Effective Probe Overhead is Minimal



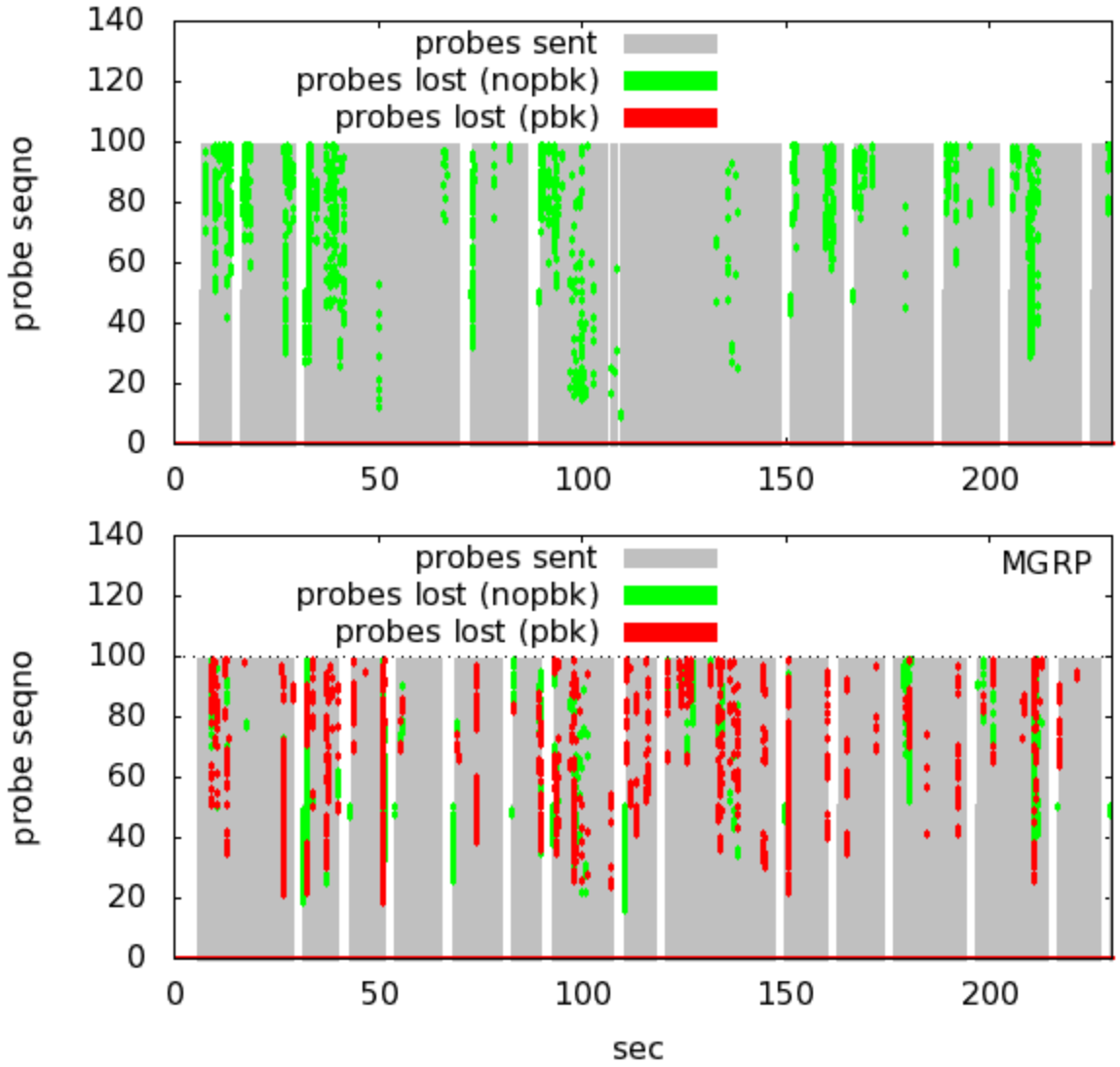
STEP: probe train pk2

Piggybacking may be too Aggressive



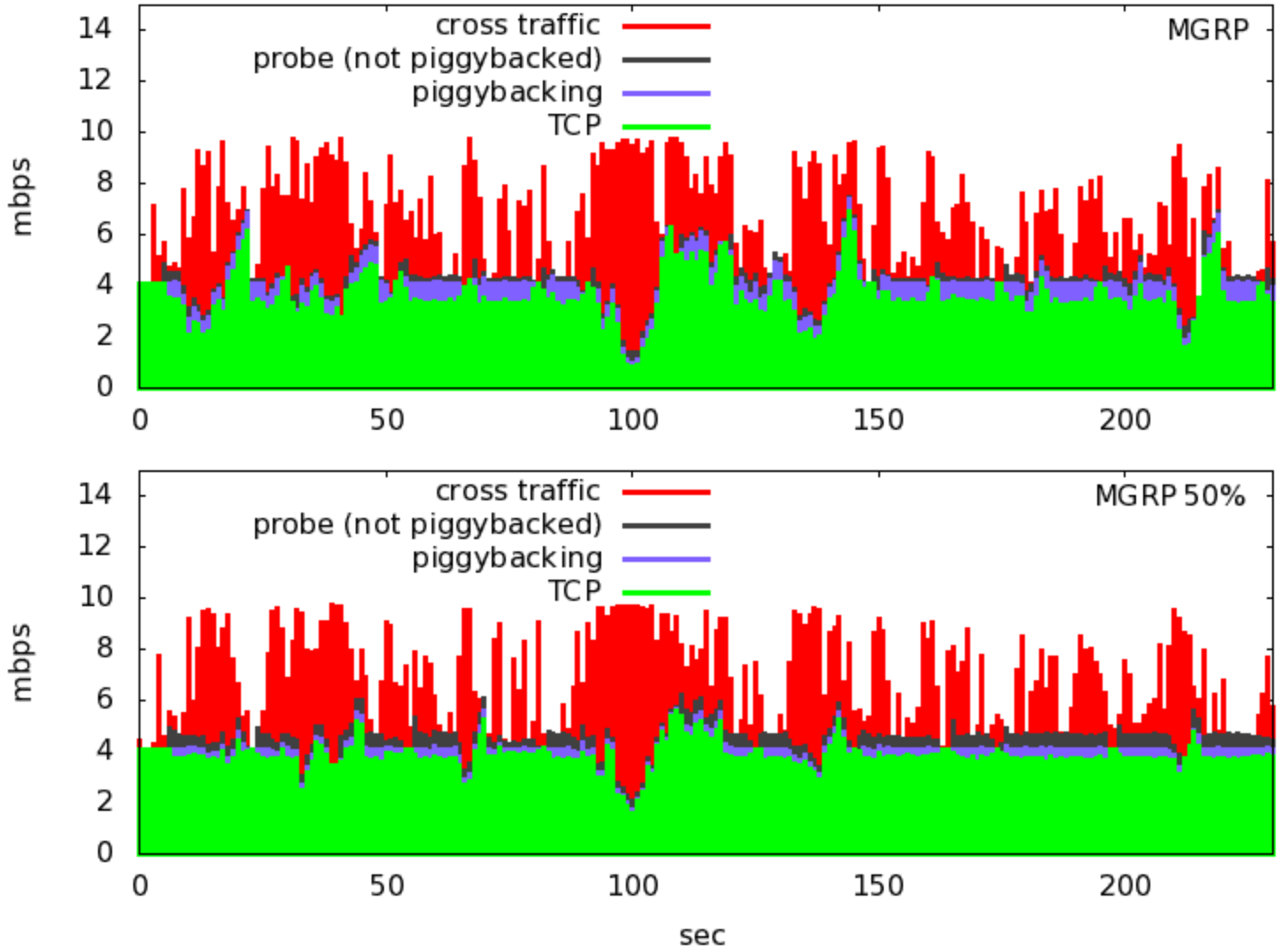
WEB: pathload pFAST

Too many piggybacked packets get lost



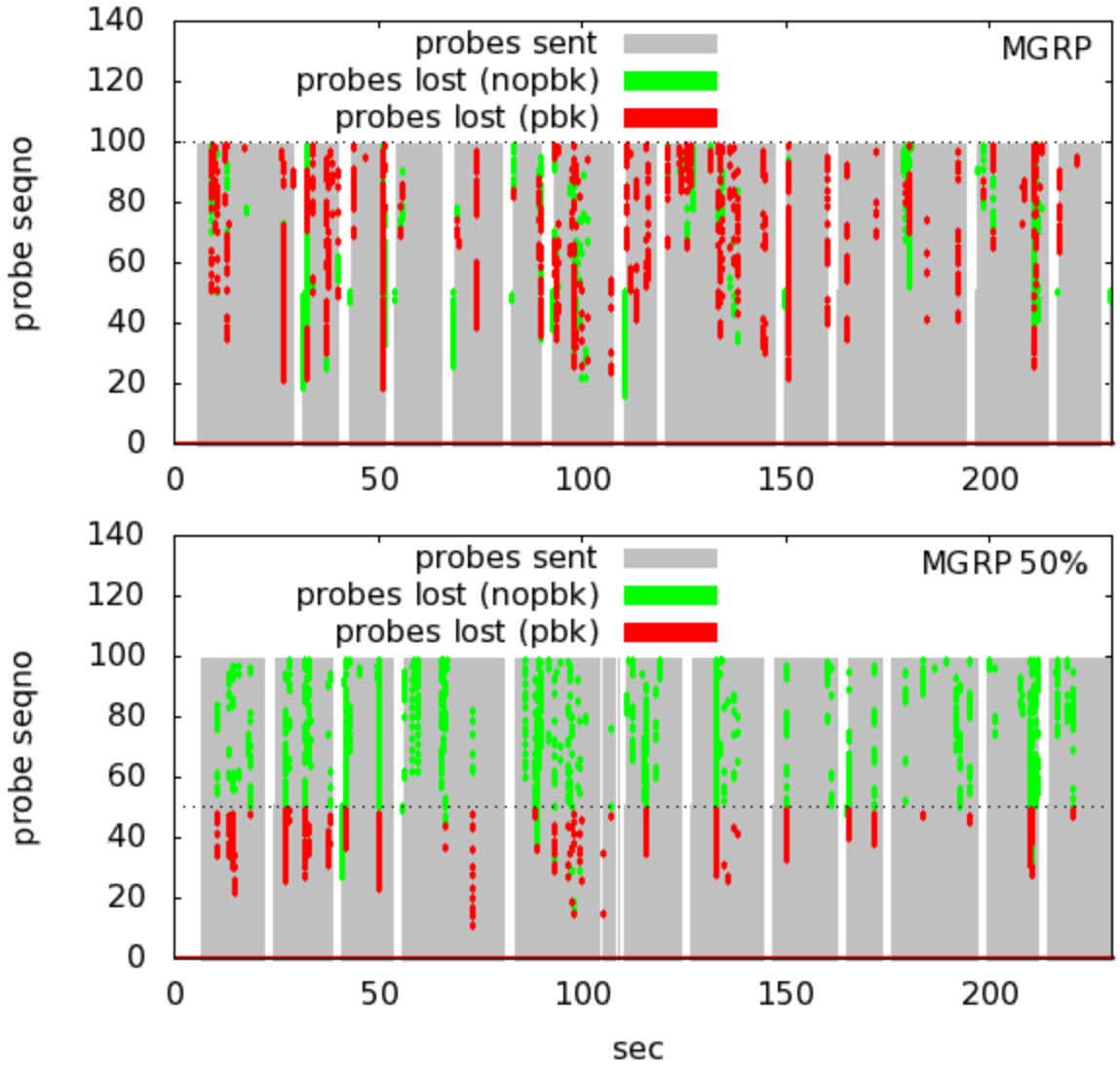
WEB: pathload pFAST

Solution: Reduce the Piggybacking Ratio



WEB: pathload pFAST

So that High Risk Probes are Avoided



WEB: pathload pFAST