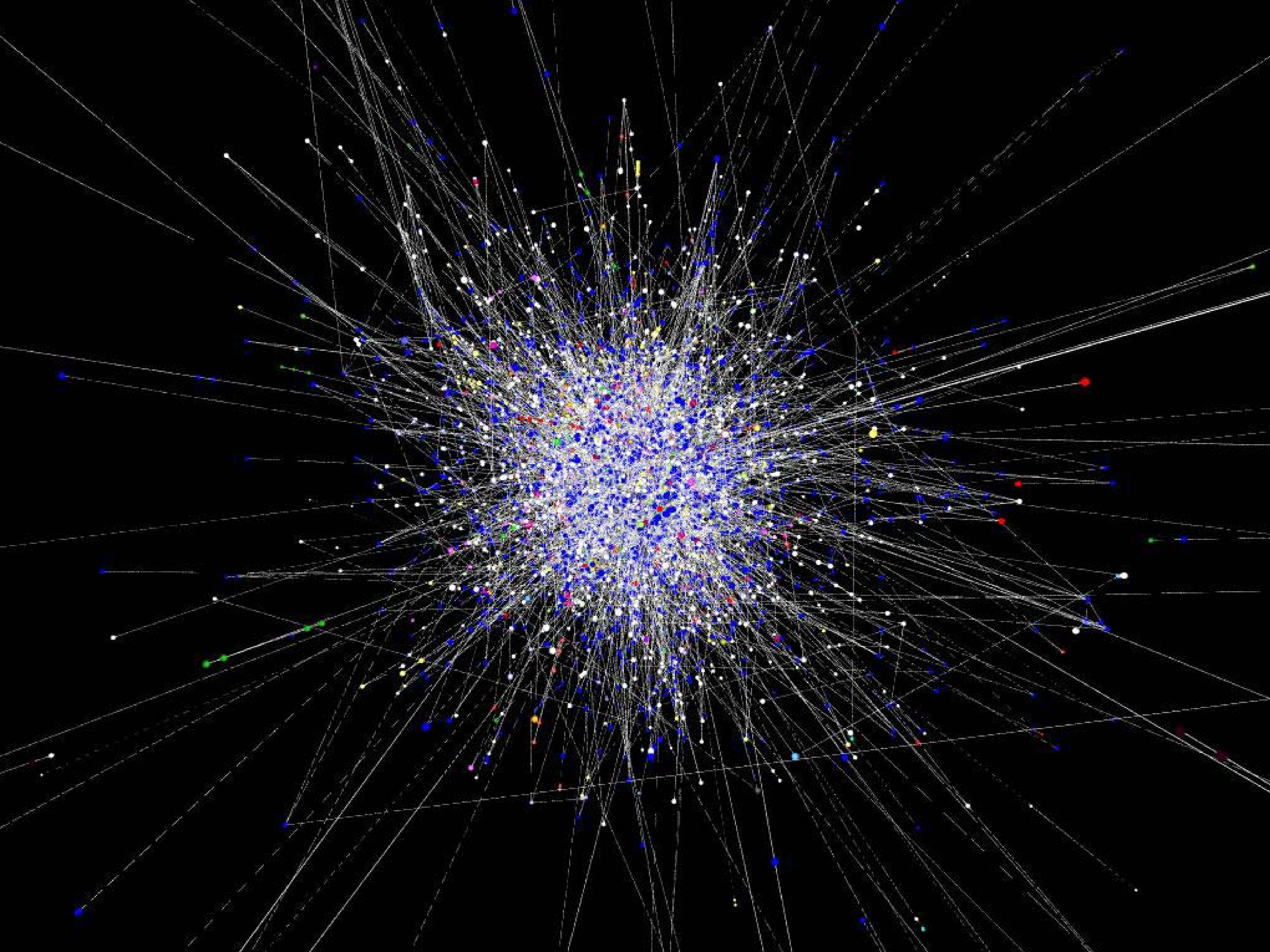
Internet Networking –

Measuring Distance and Bandwidth between hosts

Werner König

Content



I. Metrics

- Bottleneck link bandwidth
  - Higher order metrics are composed of lower order metrics
  - Bottleneck link bandwidth

    → Available bandwidth

    → TCP throughput

    → Application performance

- Distance / Latency (ping, traceroute)

I. Motivation

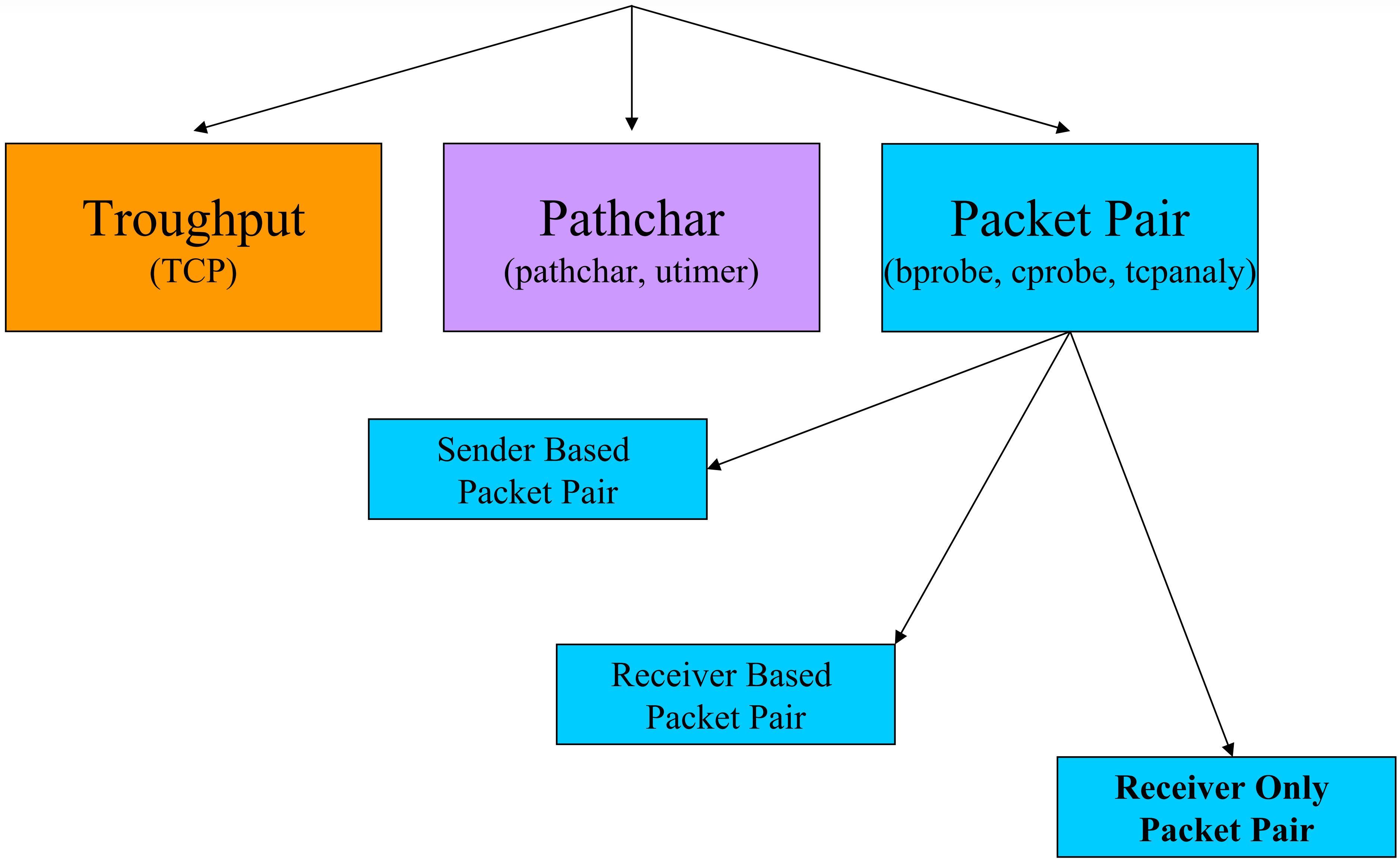
Identify network bottlenecks & high latency

→ replace them / bypass them / adapt to them.

Examples:

- Network administration
- Choose the best connection (clients, proxy)
- Choose the best network interface (Mobile Computing)
- Dynamic multicast routing trees

II. Bottleneck bandwidth algorithms



II. Troughput

Amount of data a transport protocol can transfer per unit of time

- Other metrics effects on throughput, but not on bandwidth (e.g. packet drop rate)
- Depends on application (e.g. slow CGI-script)
- Wastes network resources (e.g. TCP)
- Slowly (e.g. TCP increase slowly sending rate until one is dropped)

II. Pathchar

- Van Jacobson, Network Research Group, Lawrence Berkeley National Laboratory
- Measure round trip time
  → Software only on one host
- Varying packet sizes
  Ethernet: s = 45 sizes, 32-1500 bytes

$$s = \left\lfloor \frac{MTU}{32} \right\rfloor - 1$$

- Correlate round trip times with packet sizes to calculate bandwidth

II. Pathchar

- Active algorithm

  sends p = 32 packets/size

  ~ 1 MB/hop, 10 hops → 10 MB independent of bandwidth

  → **need high amount of bandwidth**

- Serial algorithm

  waits for acknowledgement before sending next packet

  → **slow**

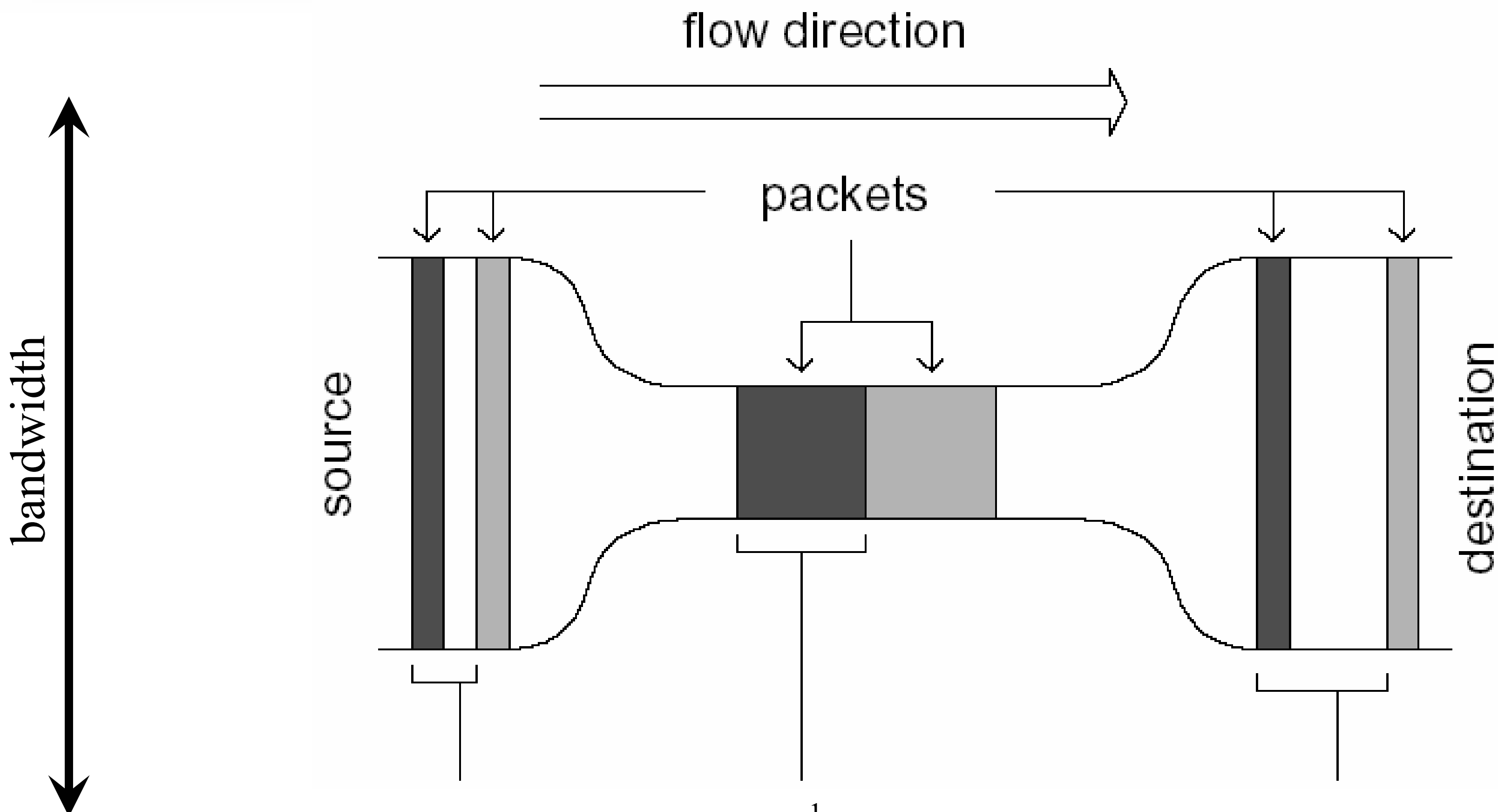
$$t = \sum_{i=1}^{h} p \bullet s \bullet l_i$$

l_i = round trip latency

h = #hops

E.g. 10 hops, Ø latency = 10ms → t = 144 sec

III. Packet Pair



$$\left(t_0^1 - t_0^0\right) < \frac{s^1}{b_{bnl}} \approx \left(t_n^1 - t_n^0\right)$$

time

III. Packet Pair

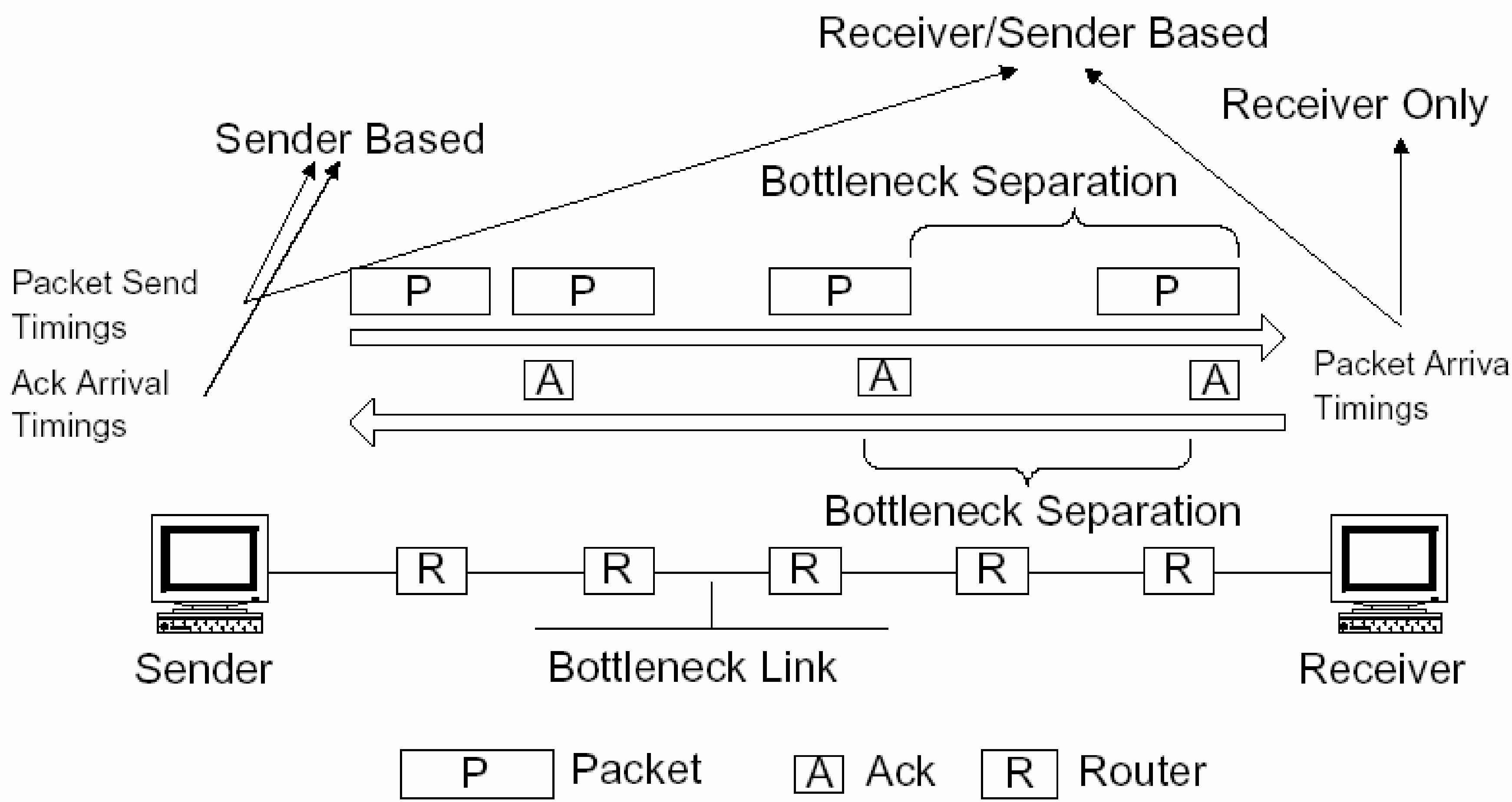
- Two Packets are queued next to each other at the bottleneck link, then they exit with

$$\Delta t = \frac{s_1}{b_{bnl}} \quad \rightarrow \text{bottleneck separation} \rightarrow \quad b_{bnl} = \frac{s_1}{\Delta t}$$

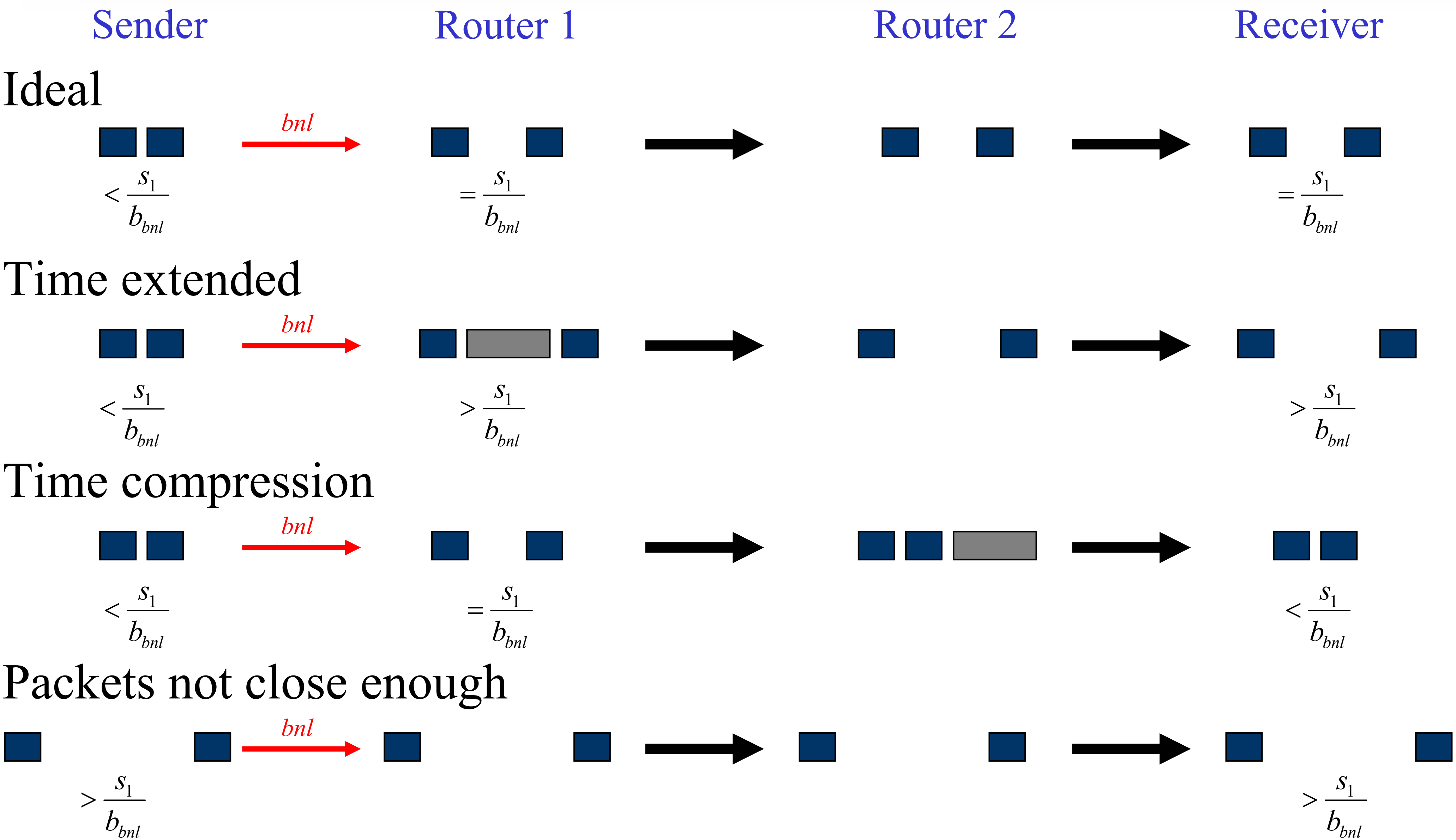
- If other packtes queue in between

$$b_{bnl} = \frac{s_1 + s_x}{\Delta t} \qquad s_x = \text{total size of other packets}$$

III. Packet Pair



III. Packet Pair



III. Packet Pair Assumptions

- For ideal results
  - Packets must be same size
  - Packet queued together at bottleneck link and traveled same path
  - FIFO, store and forward router queuing
  - No queuing downstream of bottleneck link
- Otherwise alg. produce noise
- Must filter out noise

Packet Pair Filtering

III. Packet Pair Filtering

- Filtering out noise caused by time compressed and extended packets
- Valid samples are closely clustered around the correct value
- Kernel density estimator algorithm
- Statistical valid
- Simple & fast to compute

III. Packet Pair – Pros

+ Measures true bandwidth
  - instead of throughput
+ Does not send massive amounts of data
  - unlike pathchar
+ Requires only few packet round trips
  - unlike TCP
+ Does not cause packet loss
  - unlike TCP

III. Packet Pair – Cons

- Rarely used
- Not scalable
- Slow
- Not robust on all traffic
- Not flexible to bandwidth changes
- Difficult to deploy (sender & receiver)
- Not studied under controlled condition

IV. Goals

- Make Packet Pair algorithms practical and robust enough to be widely and frequently used
- Derive simple algorithms from statistically valid network models

IV. New features

- Gradual bandwidth calculation
  - Use a packet window to measure bandwidth over varying time scales
- Receiver Only Packet Pair
  - Accuracy without deployment of special software at two nodes
- Potential Bandwidth Filtering
  - Robustly handle all packet sizes and rates

IV. Packet Window

Use *w* packets into the past to calculate the bandwidth at a particular packet

- \+ Fast, need only few packets for estimation
- \+ Agile, only recent packets are used
- \- Reduce accuracy with smaller windows

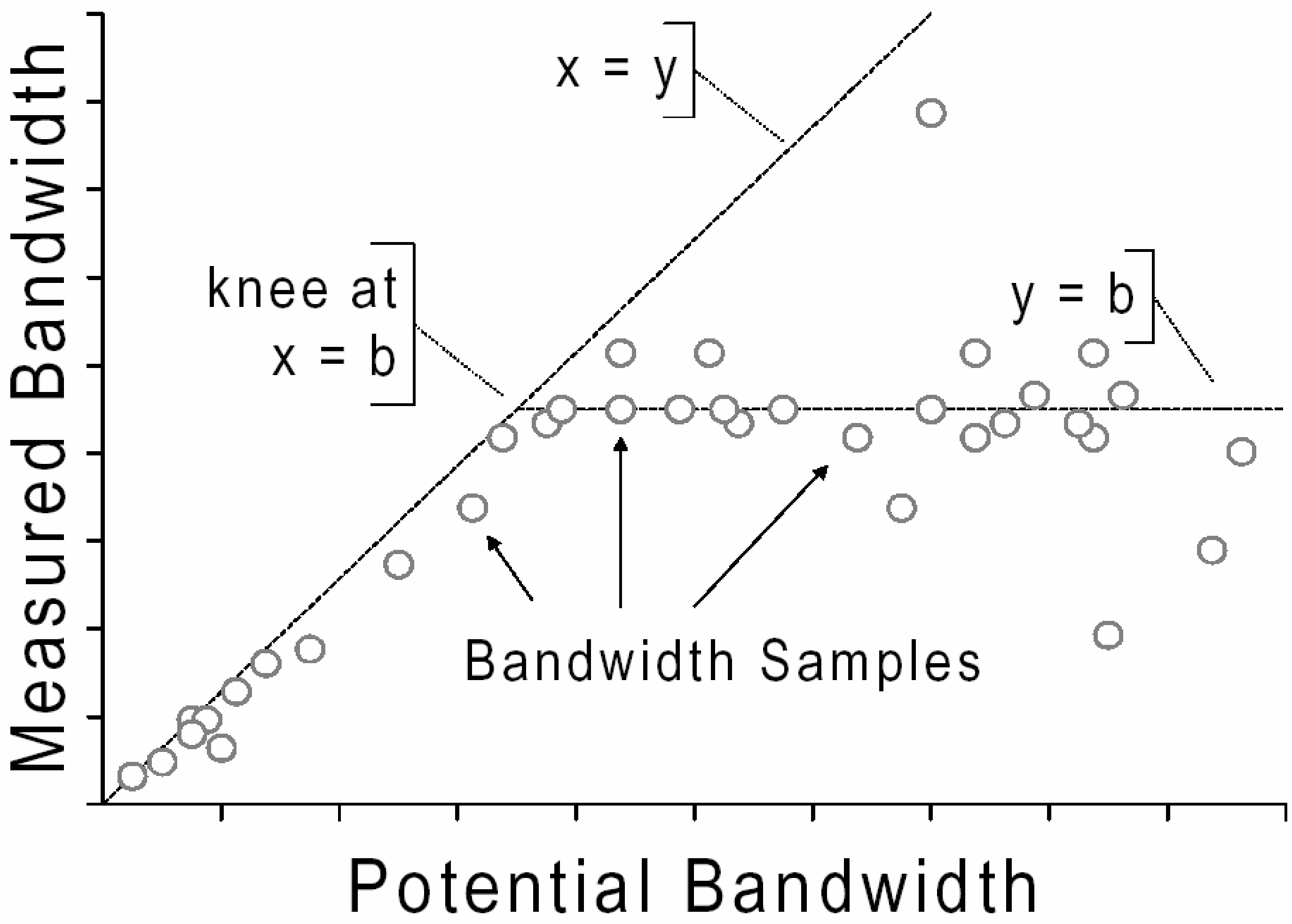
IV. Receiver Only Packet Pair

- Take measurements only at receiver
- Avoid having to deploy special software at two hosts
- Avoid inaccuracy of Sender-Based Packet Pair
- Can only measure bandwidth in the download direction

IV. Potential Bandwidth Filtering

- Problem: Existing traffic may be unsuitable for Packet Pair
  - Small or slow sent packets can mislead Packet Pair implementations
  - Exampls: TCP acknowledgements $b_{bnl} = \frac{s_1}{\Delta t}$
- Solution: Filter out small or slowly sent packets
  - PBF uses robust statistical methods to filter

IV. Potential Bandwidth Filtering



IV. Simulation Results

- Accuracy of Receiver-Only Packet Pair

Timings Taken At	Error
Sender	1200.00%
Receiver/Sender	0.09%
Receiver	0.08%

- Accuracy of Potential Bandwidth

Timings Taken At	Filtering	Bandwidth	Error
Sender	Regular	10Mb/s	44.2%
Sender	PBF	10Mb/s	7.8%
Receiver/Sender	Regular	500Kb/s	435.0%
Receiver/Sender	PBF	500Kb/s	0.0%

3/10/99

V. Conclusion

- Current bandwidth measurement techniques have several problems

- Propose statistically robust algorithms:
  - Fast estimates
  - Agile identification of bandwidth changes
  - More flexibility in deployment
  - Working with different traffic types

VI. References

Kevin Lai, Mary Baker. Measuring Bandwidth.
Department of Computer Science, Stanford University, 1999.

Kevin Lai, Mary Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. Department of Computer Science, Stanford University, 2000.

Kevin Lai, Mary Baker. Nettimer: A Tool for Measuring Bottleneck Link Bandwidth. Department of Computer Science, Stanford University, 2001.

Van Jacobson. Pathchar - a tool to infer characteristics of Internet paths. Network Research Group, Lawrence Berkeley National Laboratory, 1997

Vern Paxson. Measurements and Analysis of End-to-End Internet Dynamics. PhD thesis, University of California, Berkeley, 1997

Stephen Coast, http://www.fractalus.com/steve/stuff/ipmap.