"Unmixing" Mix Traffic or Using Blind Source Separation to Partition Flows Anonymous Communication Systems

Ye Zhu and Riccardo Bettati

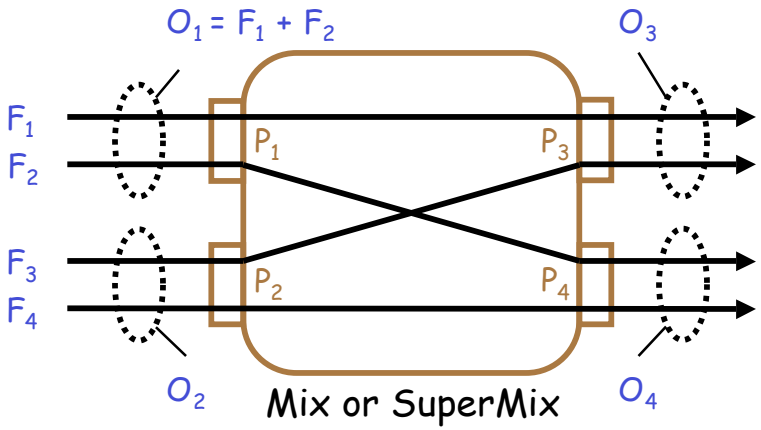
Department of Computer Science

Texas A&M University

Motivation

- Typical anonymity systems are encryption and re-routing based (e.g. Onion Routing, Tor, Freedom, TARZAN, NetCamo, ...)
- Many flow-path-reconstruction attacks to anonymity systems (Zhu et al., Levine et al., Danezis, etc.) do not scale well.
- Q: How can flow-based attacks be scaled?
- Hyp: Precondition the available traffic data.

Flow Identification vs. Flow Separation



Given:

Observations O_1, …, O_4

Flow-Path Reconstruction:

-Identify flows and paths

$F_1 : P_1 \rightarrow P_3$ and/or

$F_2 : P_1 \rightarrow P_4$ and/or

$F_3 : P_2 \rightarrow P_3$ and/or

$F_4 : P_2 \rightarrow P_4$

Flow Separation:

-Identify a group of flows, or flow aggregates, without paths

$\{ F_1 , F_2 , F_3 , F_4 \}$

Overview of Talk

- Blind Source Separation (BSS) and Independent Component Analysis (ICA)
- Flow Separation in Mix Networks
- Experimental Evaluation
  - Single-Mix case
  - Scalability of attack
  - Mix-Network case
- Conclusion / Outlook

Interlude: Blind Source Separation

- Methodology in statistical signal processing to recover source signals from a set of observed mixtures.
- Cocktail Party Problem: Extract individual's voice from mixture of voices at party.

- Let $F_1(t), \ldots, F_n(t)$ be unobserved independent "source" signals.
- Let $O_1(t), \ldots, O_n(t)$ be observation of mixtures.
- Mixing Matrix A:

$$O_i(t) = \Sigma_j \; a_{ij} \; F_j(t)$$

- Q: How to re-construct $F_j(t)$?

Blind Source Separation (II) [J.F. Cardoso 1998]

- Source Separation uses "spatial" diversity.

- "Blindness": Blind Source Separation re-constructs source signals $F_j(t)$ using only
  - observed data $O_i(t)$
  - assumption of independence among $F_j(t)$'s
  - possibly additional *a priori* information about $F_j(t)$

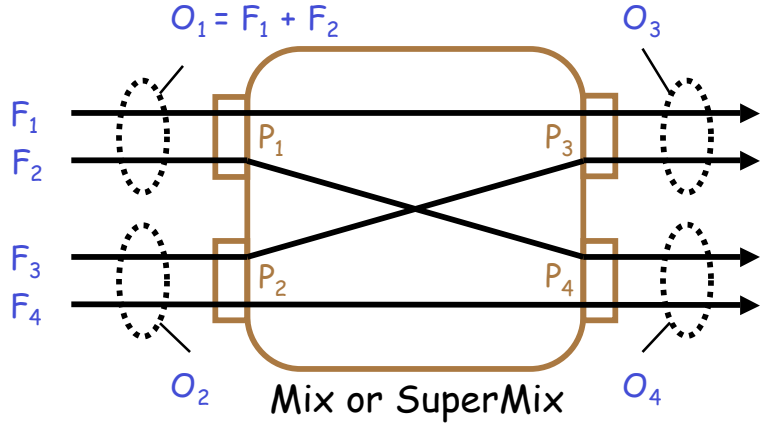
- Algorithms
  - Observation: Unless mixing matrix is non-mixing, it turns vector of independent entries into vector of non-independent entries.
  - Separation restores independence.
  - e.g. minimization of mutual information.

Blind Source Separation: Issues

- More source signals than observations (over-complete base problem)
  - Algorithms exist when number of sources is known.
  - Incomplete separation: some sources remain mixed.
- Convolutive Mixing Matrices (algorithms exist)
- Noisy observations
- Non-invertible mixing matrices
  - Row vectors of mixing matrices of MIXes are linearly dependent
  - Multicast traffic
  - Incomplete separation: some sources remain mixed.
- Estimations of separated sources are scaled and lifted.

Flow Separation as BSS Problem

- Source and observation signals as time series.
- Given $O_i = [o^i_1, o^i_2, o^i_3, \ldots, o^i_n]$ - observations of packet counts at Port P_i.
- Recover $F_j = [f^j_1, f^j_2, f^j_3, \ldots, f^j_n]$ for each flow.
- Assumptions:
  - No congestion at sender and MIX
  - Observations are synchronized



$$\begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_{j+k} \end{pmatrix} = A_{(j+k)\times m} \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{pmatrix}$$

Identifying Separated Flows: Frequency Matching

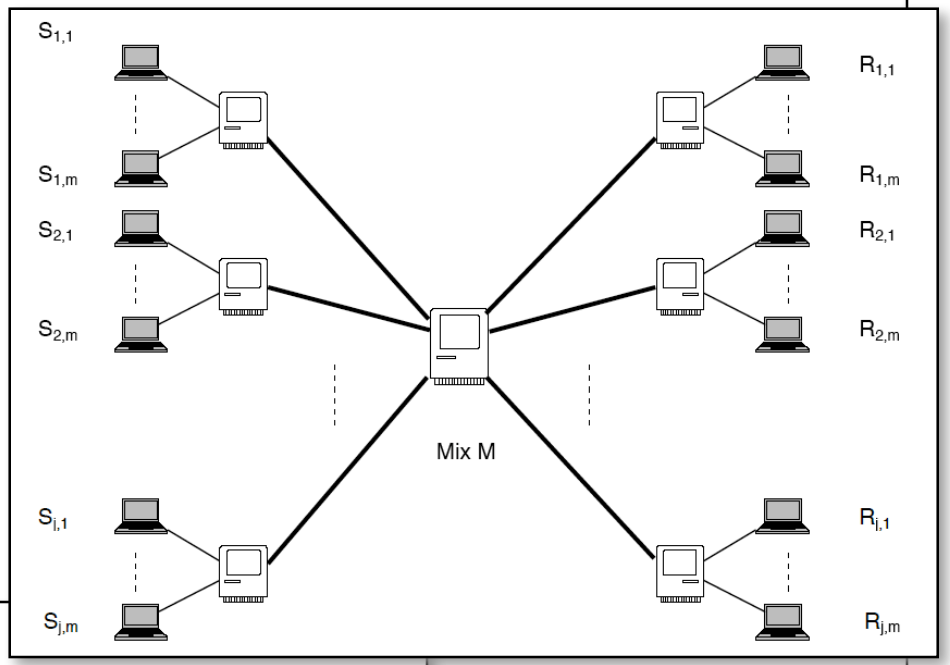
- Matching of spectrum highly effective for identifying separated flows:
  1. Captures dynamics of flows, in particular TCP.
  2. Insensitive to lifting and scaling.
  3. Effective for flow aggregates.
  4. Insensitive to congestion in network.

- Some flows are known *a priori*: Flow-path-reconstruction
- Flow characteristics unknown: Detailed traffic map across mix network can be determined.

- Metric for accuracy in experiments:
  - **Frequency Matching Rate**: Probability that separated Flow F_B matches best with actual Flow F_A.

Experimental Evaluations: Setup

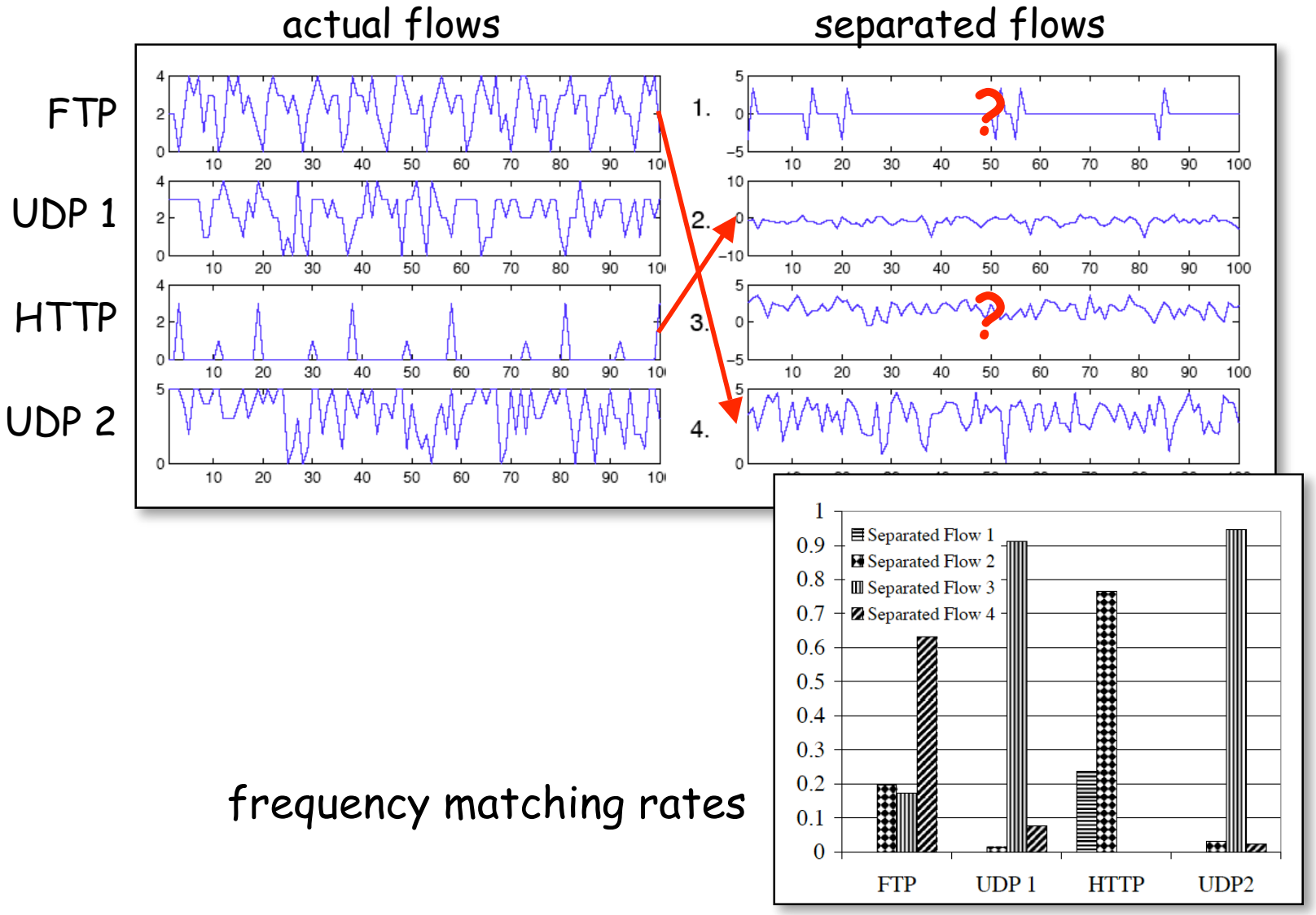
ns-2 simulations



Network:
- n x n MIX
- 10 Mbit/s link speed
- 10 ms link delay

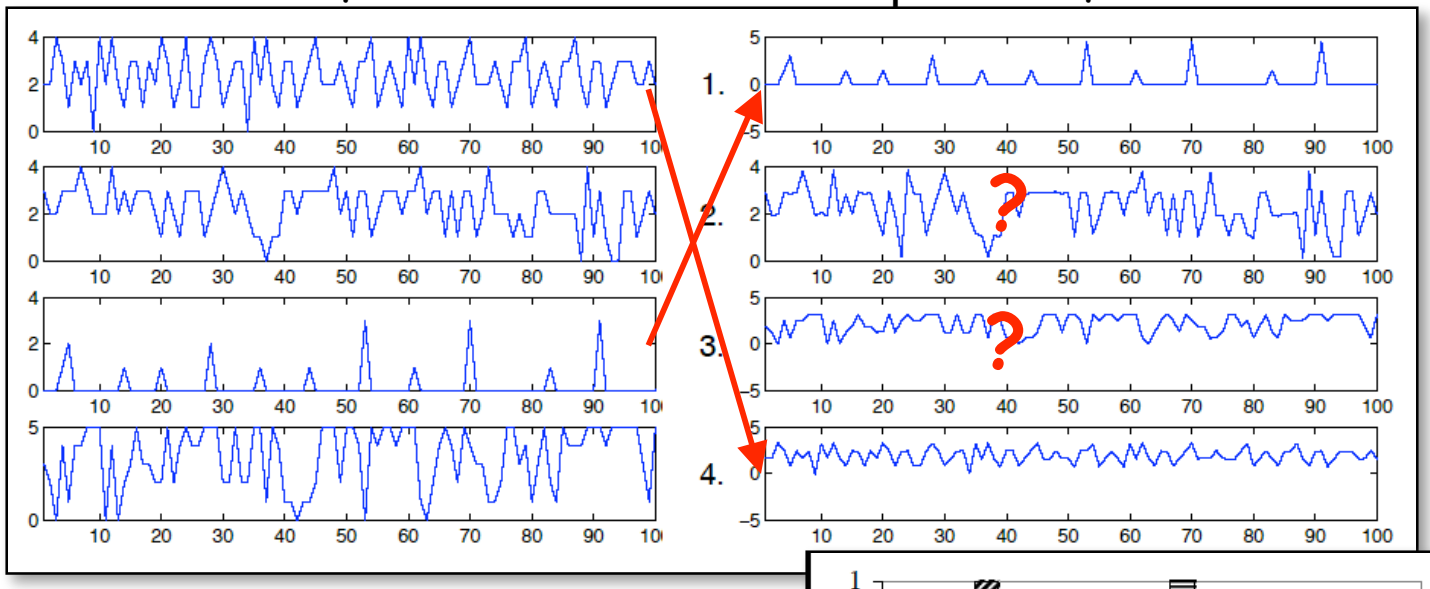
Traffic:
- FTP flow
- HTTP flows
  - avg page size 2kB
- UDP 1 flows
  - 2500 kbit/s
  - 13 ms bursts
  - 6 ms idle time
- UDP 2 flows
  - 4000 kbit/s
  - 12 ms bursts
  - 5 ms idle time
- 32 sec of traffic

2 x 2 MIX, mixed traffic, no multicast



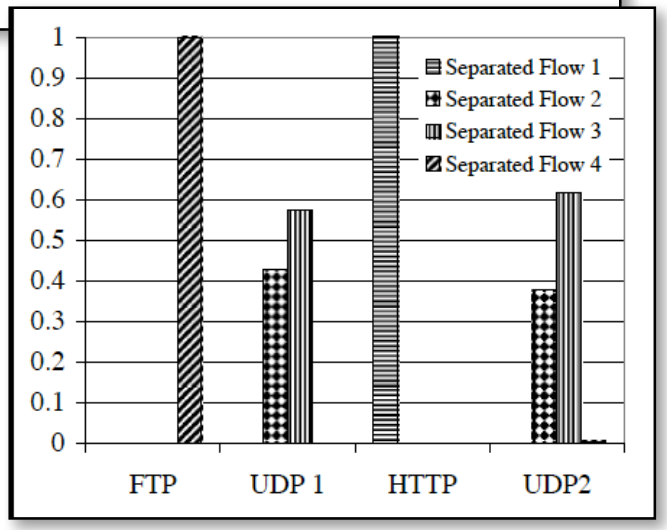
frequency matching rates

2 x 2 MIX, mixed traffic, with multicast

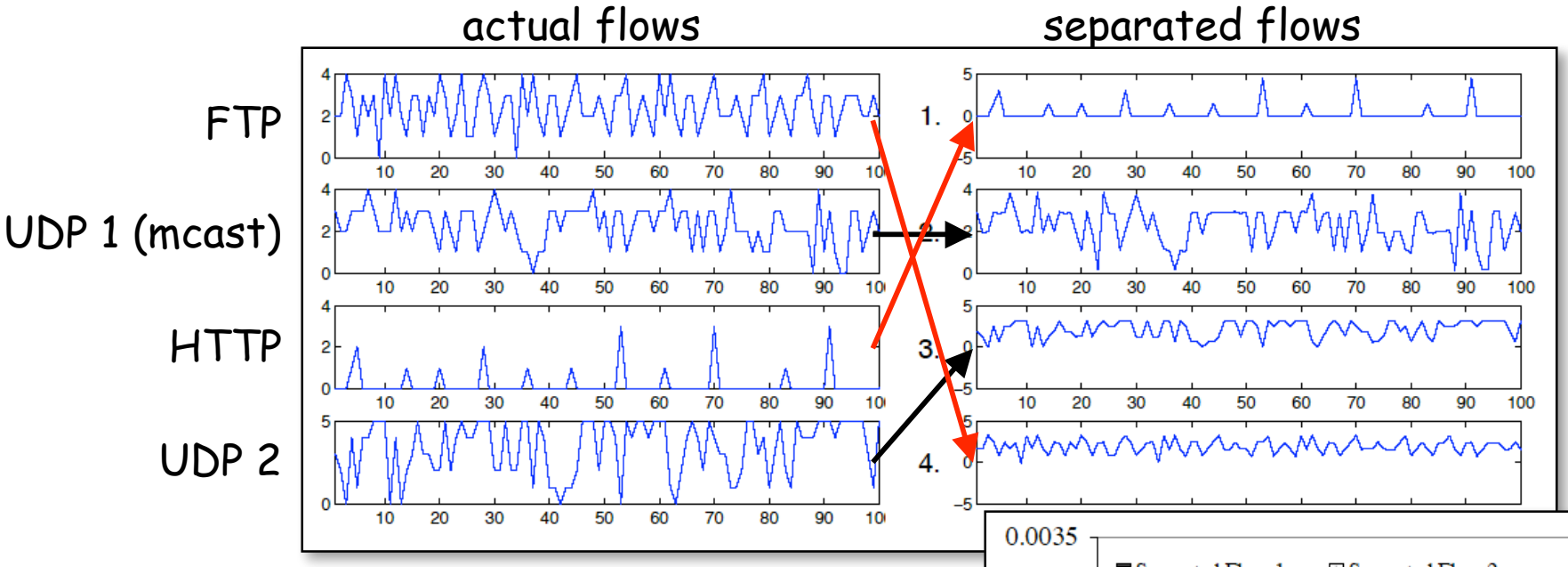


Frequency matching cannot separate the two UDP flows.

frequency matching rates

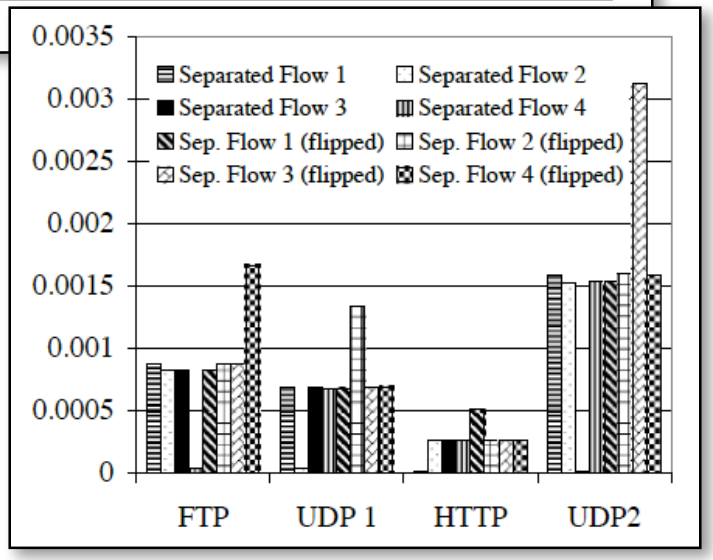


2 x 2 MIX, mixed traffic, with multicast

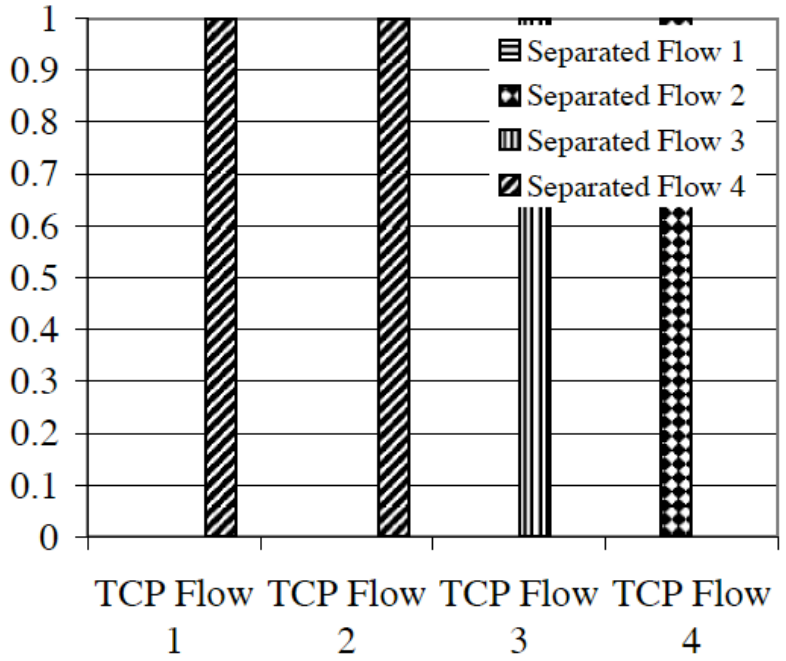


Separation works just fine, but Frequency Matching not able to correctly match UDP Flows. On the other hand, MSE matches correctly.

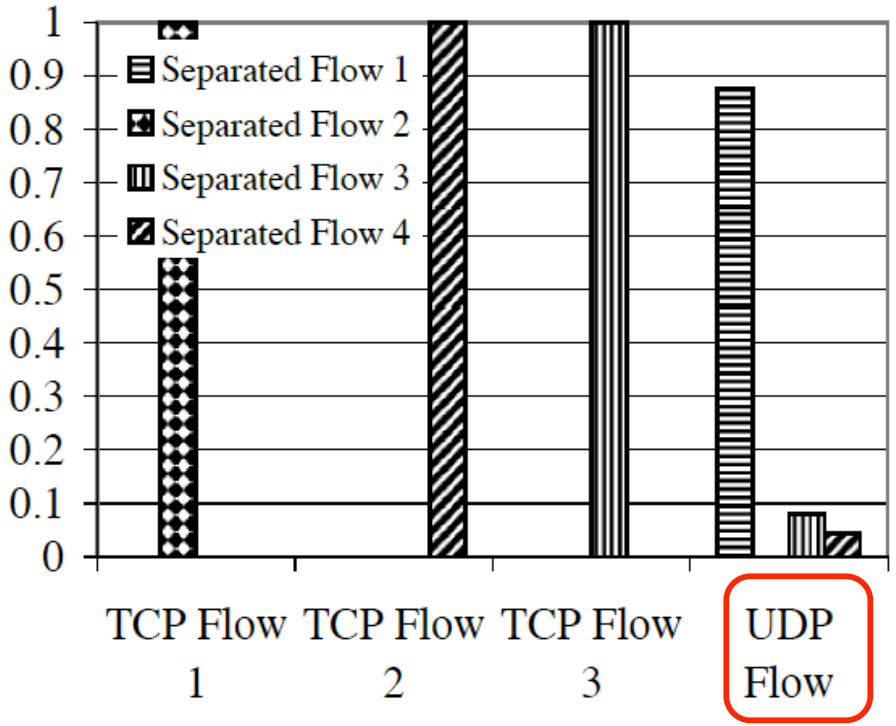
mean square error



2 x 2 MIX, TCP-only traffic

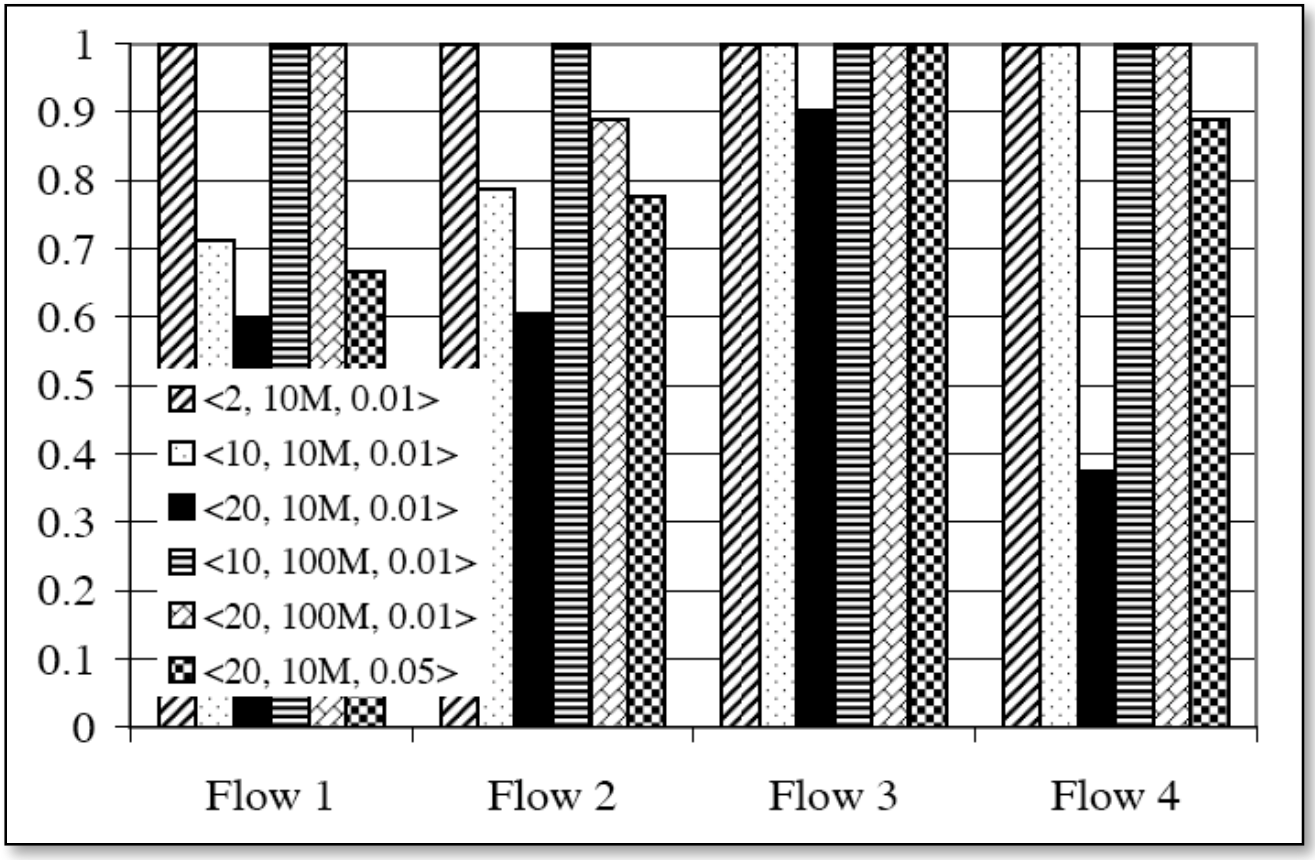


No Multicast: frequency matching



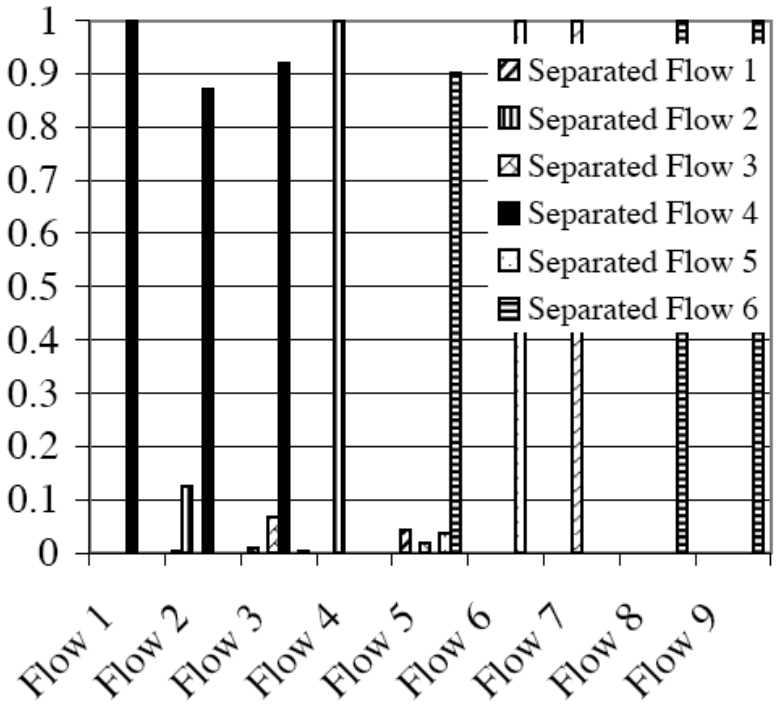
With Multicast: frequency matching

Scalability I: Congesting the (2 x 2) Mix

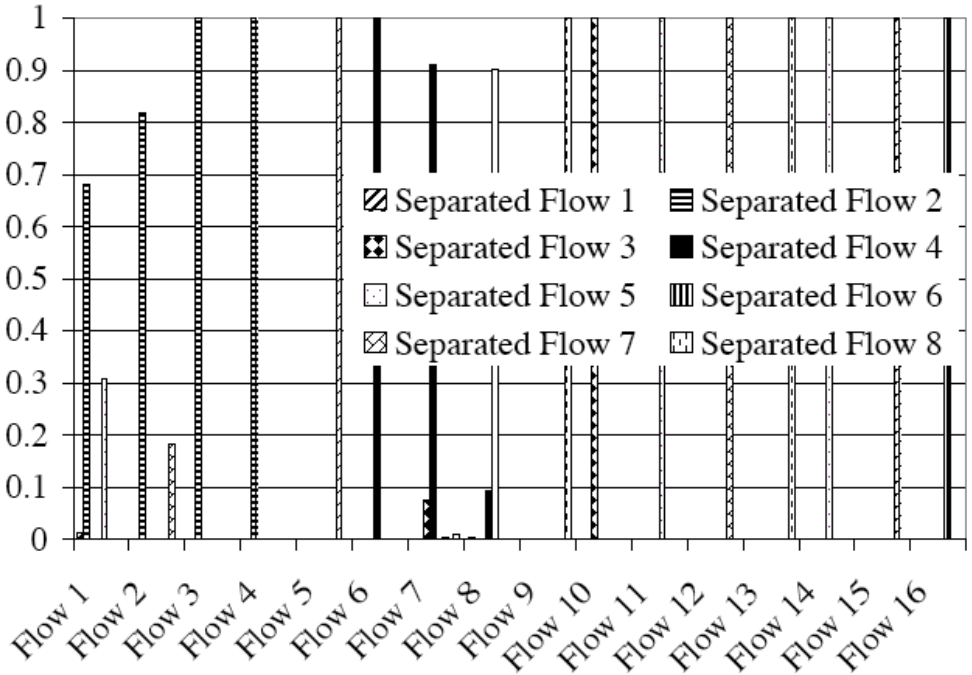


We increase the size of aggregates.

Scalability II: Larger MIX sizes

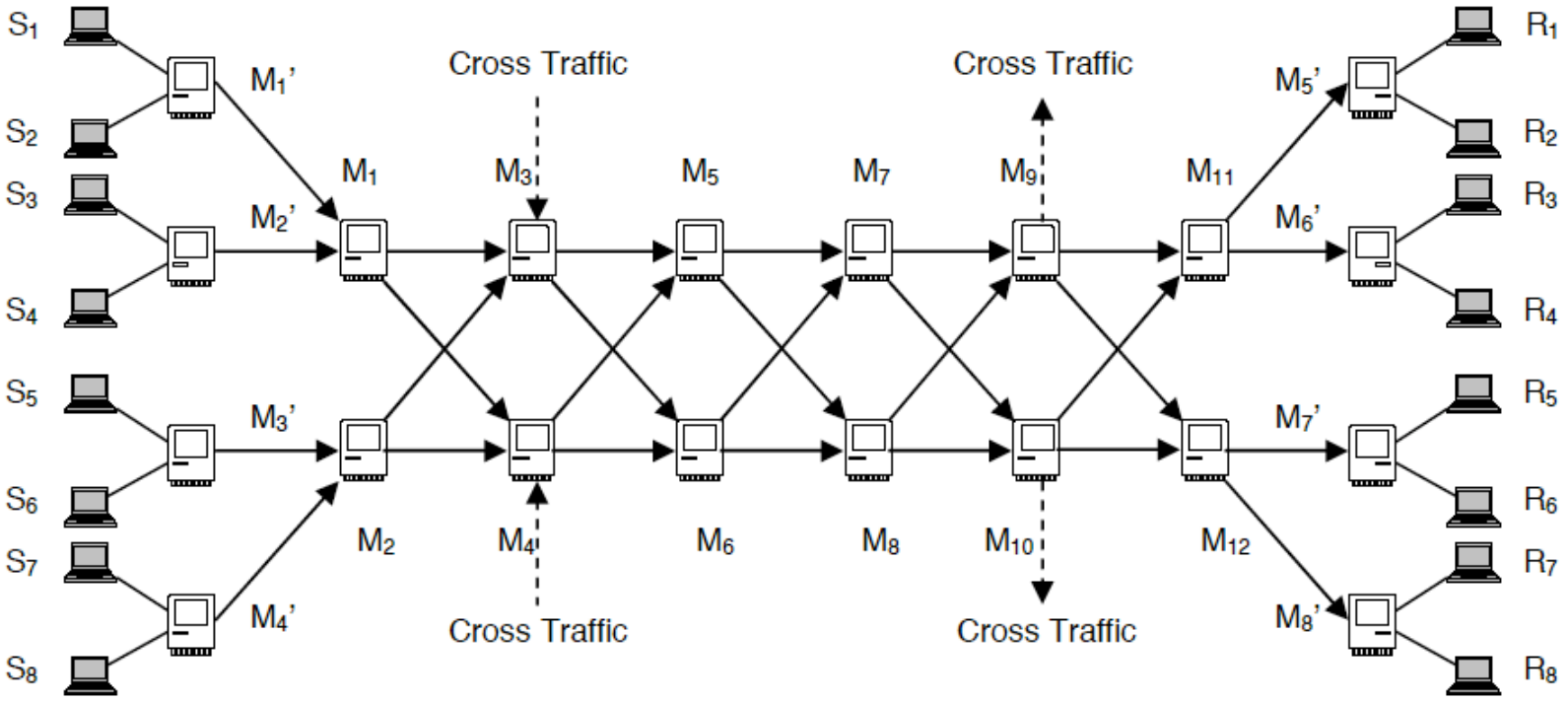


3 x 3 MIX, 9 Flows



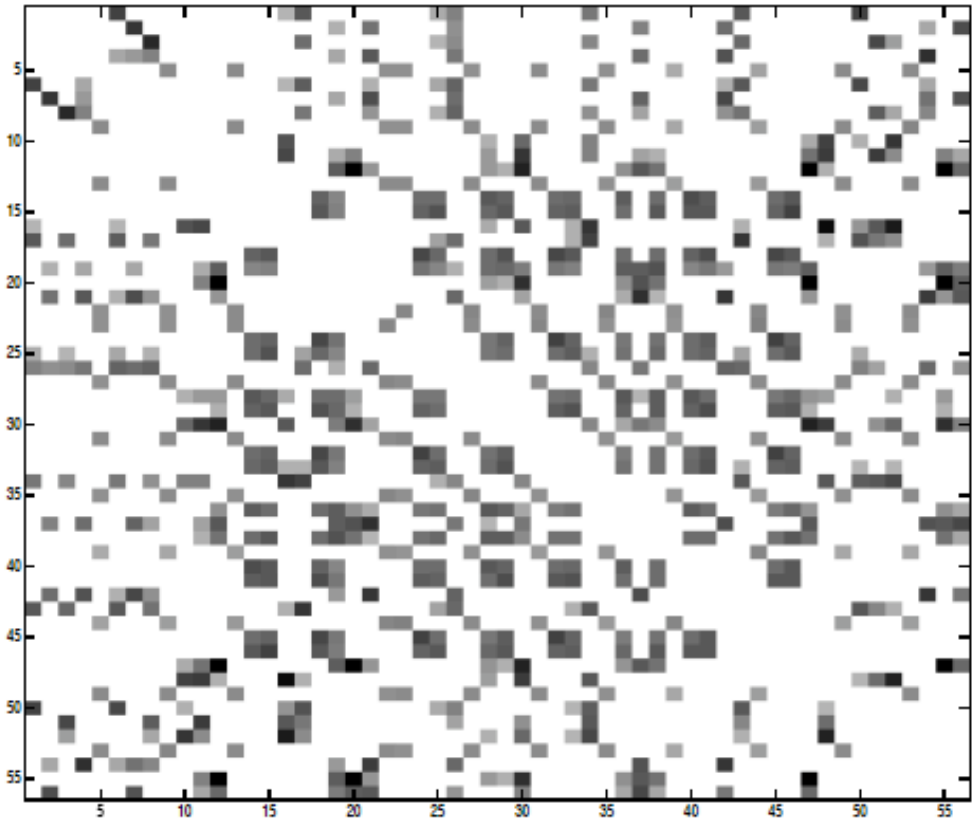
4 x 4 MIX, 16 Flows

MIX Networks



- TCP traffic
- Pareto cross-traffic

MIX Networks: Cross-Correlation Map

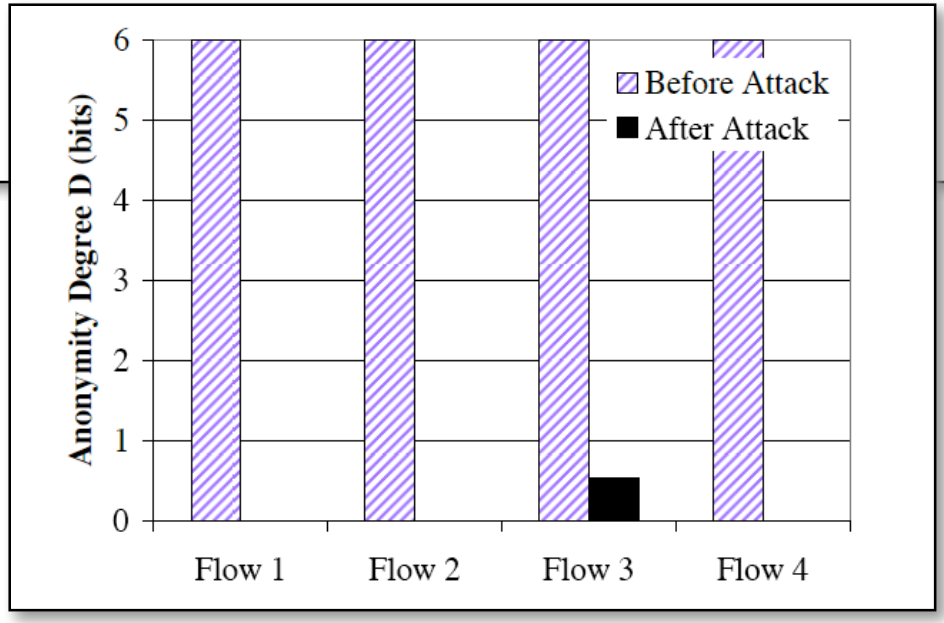


Use dynamic programming to link up separated flow aggregates to generate flow paths

MIX Networks

Anonymity Degree with *h* possible paths (Serjantov et al., Diaz et al., etc.) :

$$D = -\sum_{i=1}^{h} p_i log_2 p_i$$



Countermeasures?

1. Link padding to render observations redundant.
2. Add noise, e.g. through pool-type batching.
3. Increase dependency across flows by adding dependent dummy traffic.
4. Pad aggregate flows to render packet counts Gaussian.
   - Causes most traditional BSS algorithms to fail.
   - Does not work for newer BSS algorithms that consider time structure of signals.

Conclusion

- Flow separation as anonymity attack.
- Flow separation as preconditioner for other anonymity attacks.

- Classical example for Blind Source Separation.

- Outlook: BSS in wireless networks.
  - Traffic and power as signals.
  - Flexible placement of sensors.