Hierarchical Flow Aggregation – Problems and Open Questions

Christoph Sommer, Tobias Limmer, Falko Dressler

Computer Networks and Communication Systems,
University of Erlangen, Germany

October 2008, Munich

Outline

- Problem Statement
- Mediation Processes
- Hierarchical Aggregation
- Problems

Problem Statement

- Two Mediation Processes:
- Receive Flows
- Select Flows (Filtering)
- Create Flows (Aggregation)
- Export Flows

Problem Statement

- Two Mediation Processes:
- Receive Flows
- Select Flows (Filtering)
- Create Flows (Aggregation)
- Export Flows

- Existing Processes:
- Receive packets
- Select packets (Sampling)
- Create Flows (Metering)
- Export Flows

Flow Aggregation

Define desired Flow Keys of Compound Flows

- Defines Template to be used for export

Flow Aggregation

Define desired Flow Keys of Compound Flows

- Defines Template to be used for export

For each two incoming Flows:

- Examine these Key fields' values
- If fields have equal values: merge Flows

Flow Aggregation

Define desired Flow Keys of Compound Flows

- Defines Template to be used for export

For each two incoming Flows:

- Examine these Key fields' values
- If fields have equal values: merge Flows

Merging Flows:

- Flow Key fields' values are identical; copy values
- Non-Key fields' values differ; merge values
  - For counters, use sum
  - For ranges, use extrema
  - ...

Flow Filtering

For each incoming Flow:

- Compare Flow's fields with series of filtering criteria
- Accept if all fields have equal values (logical AND)
- Identical to PSAMP "match" algorithm

Flow Filtering

For each incoming Flow:

- Compare Flow's fields with series of filtering criteria
- Accept if all fields have equal values (logical AND)
- Identical to PSAMP "match" algorithm

Additional Filters desirable:

- Check non-equality of fields values (logical NOT)
- Accept based on list of possible values (logical OR)
- Accept based on range of possible values

Architectural Considerations

What to do first – filtering or aggregation?

- Both options make sense:
  - Aggregating filtered flows
  - Filtering aggregated flows
- Even more options in hierarchical deployment:
  - Filter, aggregate, (export, collect,) aggregate, filter

Architectural Considerations

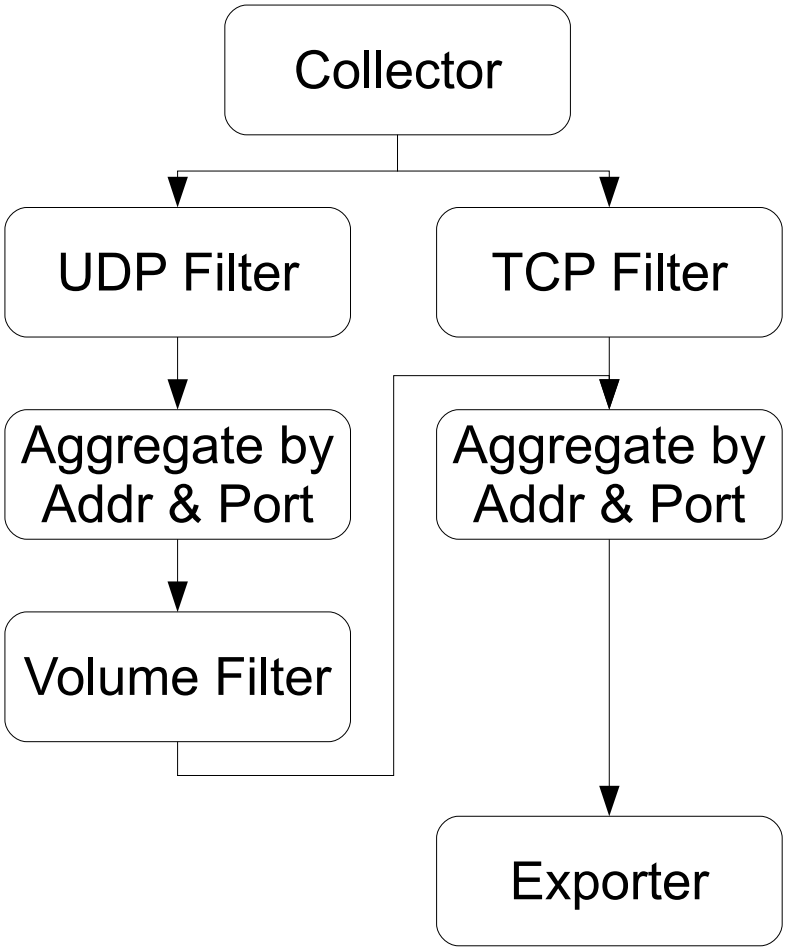
What to do first – filtering or aggregation?

- Both options make sense:
  - Aggregating filtered flows
  - Filtering aggregated flows
- Even more options in hierarchical deployment:
  - Filter, aggregate, (export, collect,) aggregate, filter

Need flexible way of configuring process order

- Best order of processing steps varies widely

Flow Processing Example



- In:
  1x UDP 1.2.3.4:80 > 1.2.3.5:1080
  1x UDP 1.2.3.4:80 > 1.2.3.5:1080
  1x TCP 1.2.3.4:80 > 1.2.3.5:1080
  1x UDP 2.3.4.5:80 > 2.3.4.6:1080

- Out:
  3x 1.2.3.4:80 > 1.2.3.5:1080

Communicating Flow Treatment

How PSAMP communicates selection criteria:

- Track which Selectors a packet has passed
- Assign identifier to this Selection Sequence
- Annotate created Flow with this identifier

Communicating Flow Treatment

How PSAMP communicates selection criteria:

- Track which Selectors a packet has passed
- Assign identifier to this Selection Sequence
- Annotate created Flow with this identifier

Use this for communicating Flow treatment?

- Track which Filters and Aggregators a Flow passed
- Assign identifier to this sequence of processes
- Annotate Compound Flow with this identifier

Problems

Need to include information about multiple branches

- When merging Flows from multiple Filters:
- Flows can pass either one to be merged
- Just refer to all contributing processes' IDs?

Problems

Need to include information about multiple branches

- When merging Flows from multiple Filters:
- Flows can pass either one to be merged
- Just refer to all contributing processes' IDs?

Should later Filters be able to consider earlier Filters' criteria?

- e.g. filter for "TCP or UDP" and discard the protocol;
  Can one still expect this to match Filter "not ICMP"?
- Filters then need to interpret configuration of earlier ones

Problems

Need to include information about multiple branches

- When merging Flows from multiple Filters:
- Flows can pass either one to be merged
- Just refer to all contributing processes' IDs?

Should later Filters be able to consider earlier Filters' criteria?

- e.g. filter for "TCP or UDP" and discard the protocol;
  Can one still expect this to match Filter "not ICMP"?
- Filters then need to interpret configuration of earlier ones

How to configure these Mediators?

- Structure is no longer a tree...

Hierarchical Flow Aggregation – Problems and Open Questions

Christoph Sommer, Tobias Limmer, Falko Dressler

Thanks!

christoph.sommer@informatik.uni-erlangen.de