

SCRIPT Public Workshop
January 20, 2010, Zurich, Switzerland

SCRIPT: An Architecture for IPFIX Data Distribution

Peter Racz

Communication Systems Group CSG
Department of Informatics IFI
University of Zürich UZH



Outline

- ❑ Motivation
- ❑ Proposed Solution
- ❑ SCRIPT Application Scenarios
- ❑ SCRIPT Architecture
 - SCRIPT Node
 - SCRIPT Controller
- ❑ Deployment

Motivation

- ❑ Situation
 - Increasing **link speed** and **network traffic**
 - More data to be processed by an analysis application
 - Analysis **application requirements**
 - Correlation of flow records from multiple exporters

- ❑ Problem
 - **Scalability** of a centralized collector

- ❑ Current approach
 - **Packet sampling** and **flow sampling**
 - Measurement inaccuracy
 - **Dedicated hardware**
 - Higher costs
 - **Replace hardware** if performance is not enough
 - Higher costs

Proposed Solution

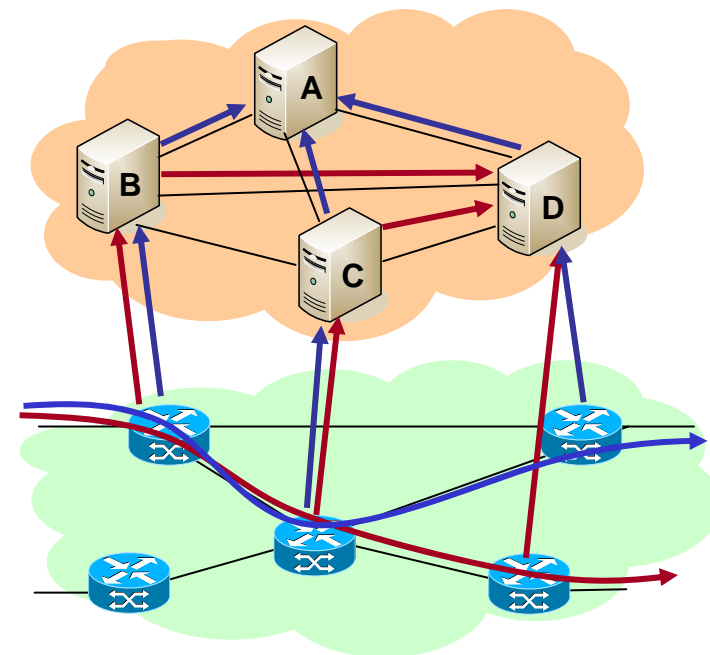
- ❑ **Distribute traffic data** to multiple traffic analyzers
 - Make use of the resources of multiple nodes
 - Reduce the amount of data to be processed on a single node
 - Increase the time available to process a single flow record
 - Trade-off
 - More network traffic
- ❑ **SCRIPT**
 - Distributed platform for IP flow accounting and analysis
 - Based on NetFlow v9 / IPFIX records
- ❑ **Goals**
 - Increase the amount of data that can be analyzed
 - Dynamic configuration (add/remove resources)
 - Avoid single point of failures
 - Provide load-balancing of workload
 - No dedicated hardware

Scenario 1: Flow Record Storage

- ❑ SCRIPT platform is used to store flow records
 - To increase the storage capacity
 - To balance the usage of storage capacities of SCRIPT nodes
 - To achieve redundant storage
 - To reduce query time of stored flow records

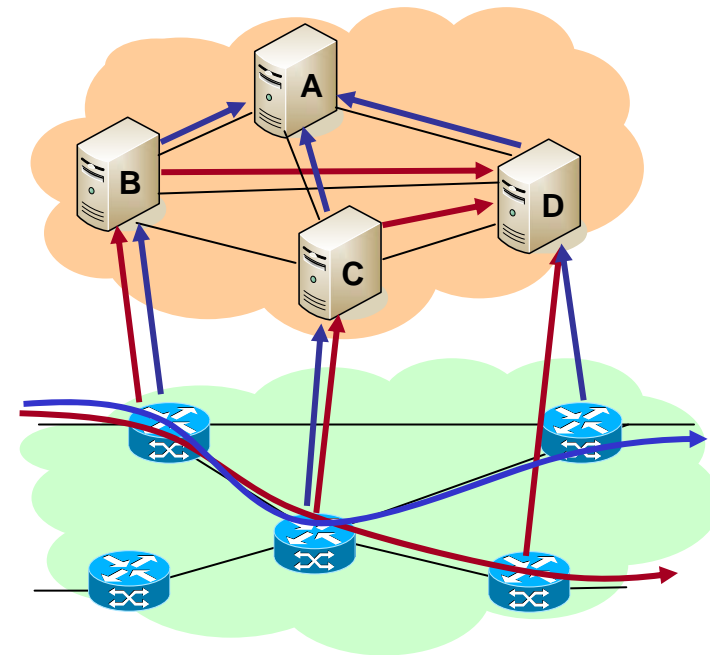
- ❑ Flow record exporting
 - Record can contain any attribute

- ❑ Flow record routing
 - Routing based on different attributes
 - Flow-keys-based distribution (e.g., 5-tuple)
 - Exporter-based distribution
 - Template-based distribution



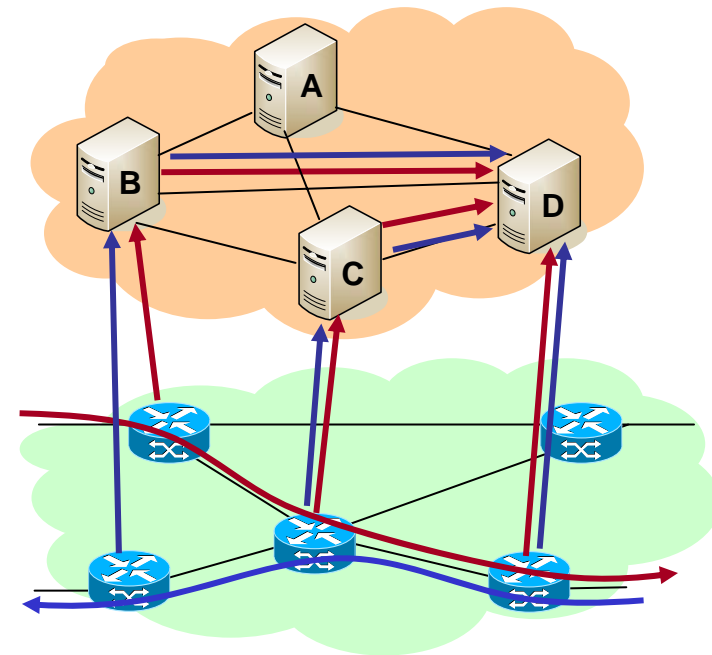
Scenario 2: One-Way Delay Measurement

- ❑ Measure one-way delay based on flow records
 - To distribute the processing overhead among SCRIPT nodes
 - To perform delay calculation in near real-time
- ❑ Flow record exporting
 - Flow records for single packets
 - Sampling function such that if a packet is selected by one exporter, then it will be selected by all exporters on the path
 - Flow record has to include
 - 5-tuple
 - Timestamp of the flow creation
 - Origin exporter
 - Hash value calculated on (part of) the payload
 - Clocks have to be synchronized
- ❑ Flow record routing
 - Flow records generated for the same packet but exported by different routers are routed to the same SCRIPT node



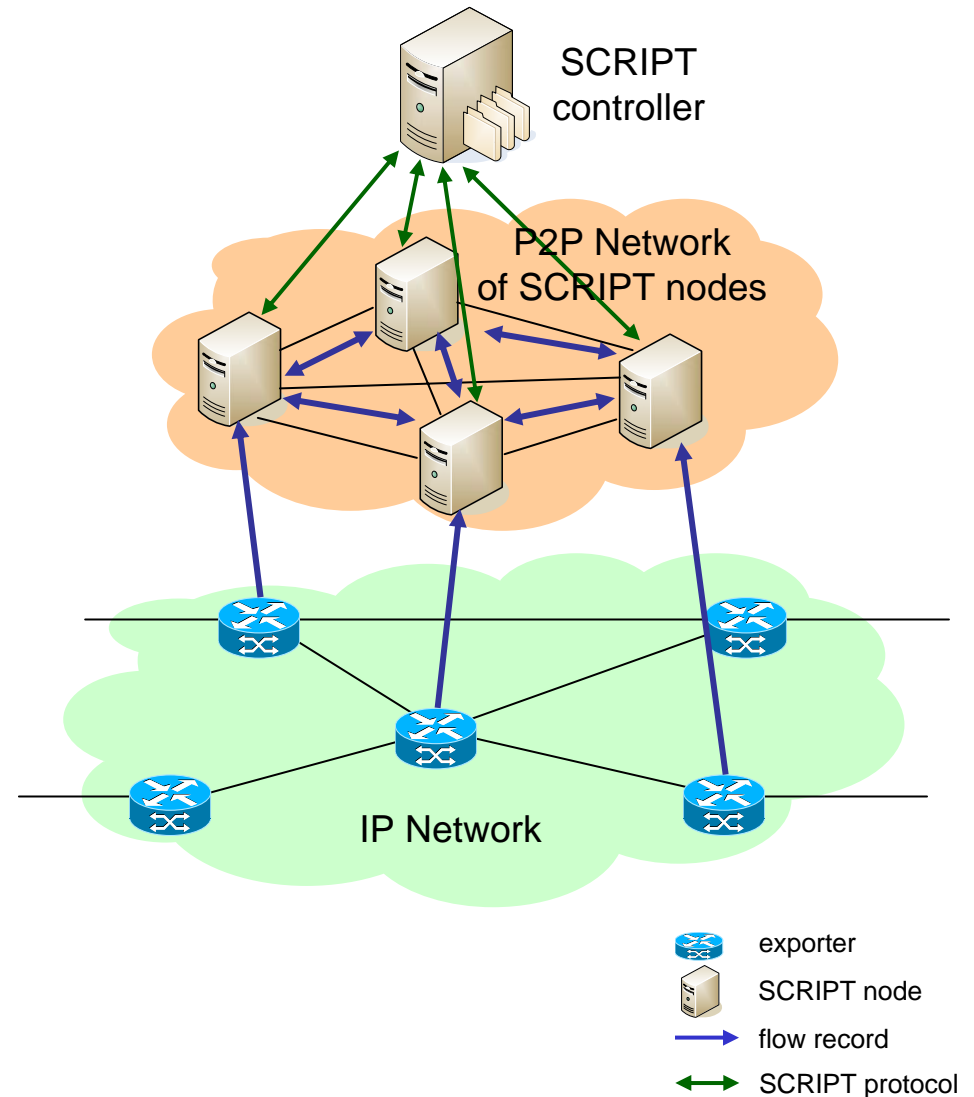
Scenario 3: Asymmetric Route Detection

- ❑ Detect asymmetric routes based on flow records
 - To distribute the processing overhead among SCRIPT nodes
 - To detect asymmetric routes in near real time
- ❑ Flow record exporting
 - Flow records have to include
 - 5-tuple
 - Timestamp of the first packet
 - Origin exporter
 - Clocks have to be synchronized
- ❑ Flow record routing
 - Flow records belonging to the same flow (in both directions) but exported by different routers are routed to the same SCRIPT node.

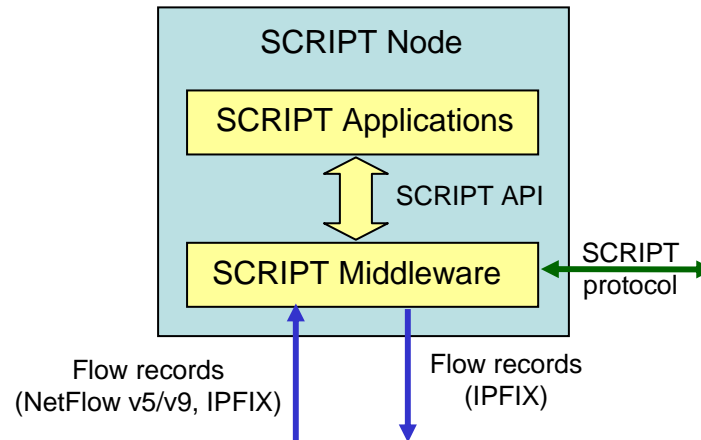


SCRIPT Architecture

- ❑ P2P-based flow record routing overlay
 - Built by SCRIPT nodes
 - Receive flow records from routers
 - Distribute flow records to SCRIPT nodes based on IPFIX
- ❑ Supports
 - Flexible flow record routing policies
 - Different analysis applications
 - Load distribution and scalability
- ❑ Exporters
 - Have one or more SCRIPT nodes registered as flow collectors
 - Can switch between SCRIPT nodes
 - In case of failure
 - In case of overload of a single SCRIPT node



SCRIPT Node



□ SCRIPT node

- A logical entity that runs the SCRIPT middleware and one or more SCRIPT applications
- Receive flow records from exporters (do not generate flow records)
- Can forward flow records to other SCRIPT nodes (flow record routing)

□ SCRIPT middleware

- A common communication and coordination layer between SCRIPT nodes
- Provides common functionality for applications
 - Management and coordination of SCRIPT nodes, P2P overlay management (join/leave)
 - Flow collecting, exporting processes, flow record routing, mediation function
 - Delivery of flow records to applications

□ SCRIPT application

- Implements a certain traffic analysis application based on flow records
- On top of the middleware using the SCRIPT API

SCRIPT Controller

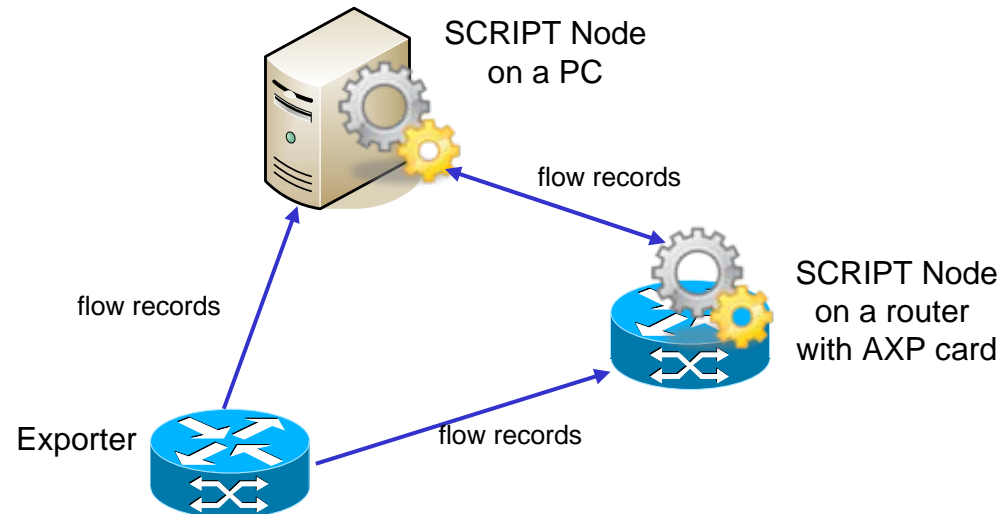
- A central entity responsible for **management tasks**
 - SCRIPT node identity assignment
 - SCRIPT node bootstrapping
 - Flow template management and synchronization

- **Does not participate** in flow record routing

Template Synchronization

- ❑ Same template definition may carry **different template IDs**
 - e.g. when exported by different routers, or after a router reboot
- ❑ SCRIPT matches different template IDs of the same template definition to a **global template ID**
- ❑ Mechanism:
 - SCRIPT controller maintains a database of template definitions and their global template IDs
 - Each SCRIPT node locally maintains such a database which is updated in time
 - When receiving a template definition, if it is a new one, it is sent to the Controller
 - The controller searches the template database and returns a corresponding global template ID
 - The SCRIPT node locally maps (exporterID, templateID) → global template ID
 - The template ID for each received record is set to the corresponding global template ID

Deployment



- ❑ Exporters
 - Create and export flow records to SCRIPT nodes
 - Using IPFIX, NetFlow v5, or NetFlow v9
- ❑ SCRIPT nodes can be deployed on
 - Linux servers
 - AXP cards installed in a Cisco router

Thank you for your attention!