Revisiting Interdomain Root Cause Analysis from multiple vantage points

Anthony Lambert
anthony.lambert@orange-ftgroup.com

Mickael Meulle
michael.meulle@orange-ftgroup.com

Jean-Luc Lutton
jeanluc.lutton@orange-ftgroup.com

Diffusion Libre

Septembre 2005

Overview

- BGP is the glue of the Internet, so any interdomain routing change threatens end-to-end performances and reachability.

- A tool that would detect and localize accurately events originating interdomain routing changes as early as they occur is thus of great interest:
  - Increased reactivity of NOCs during outages
  - Increased proactivity of NOCs detecting small recurrent events for instance
  - Better peering decisions identifying which peer is reliable or not

Principles

- Based on the analysis of BGP updates collected at different vantage points, we aim at localizing which ASs are responsible for the routing changes observed.

- More precisely, the steps interdomain Root Cause Analysis is intended to perform are:
  - To decompose the stream of updates collected at different monitoring points into clusters so that each cluster contains all the BGP updates related to the same event.
  - To localize inside each cluster the AS(s) that are the more likely to have originated the underlying event.

Challenges

- Given a set of monitored routers in various ASs:
  - We only perceive some lessened effects of events: the spread of updates depends on the sum of local decisions of the routers from the event location to the monitoring points.
  - The effects of an event spread over time: because of MRAI timers for instance.
  - Effects of different events overlap: events can happen close in time.
  - Information collected at monitoring points is biased: the nearer an AS is from a monitoring point, the more likely it is to appear in updates.

Previous Approaches

- Decomposing the stream of updates
  - Use of timers based on the arrival frequency of updates
  - Can use some level of aggregation

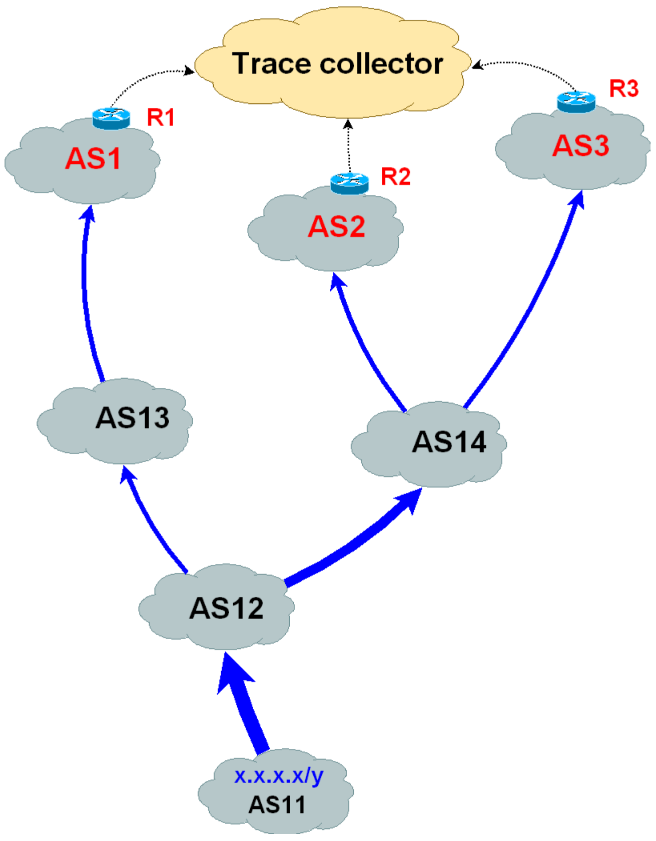
- Localization of ASs responsible for events
  - Events originators are supposed to be either on old or new best paths

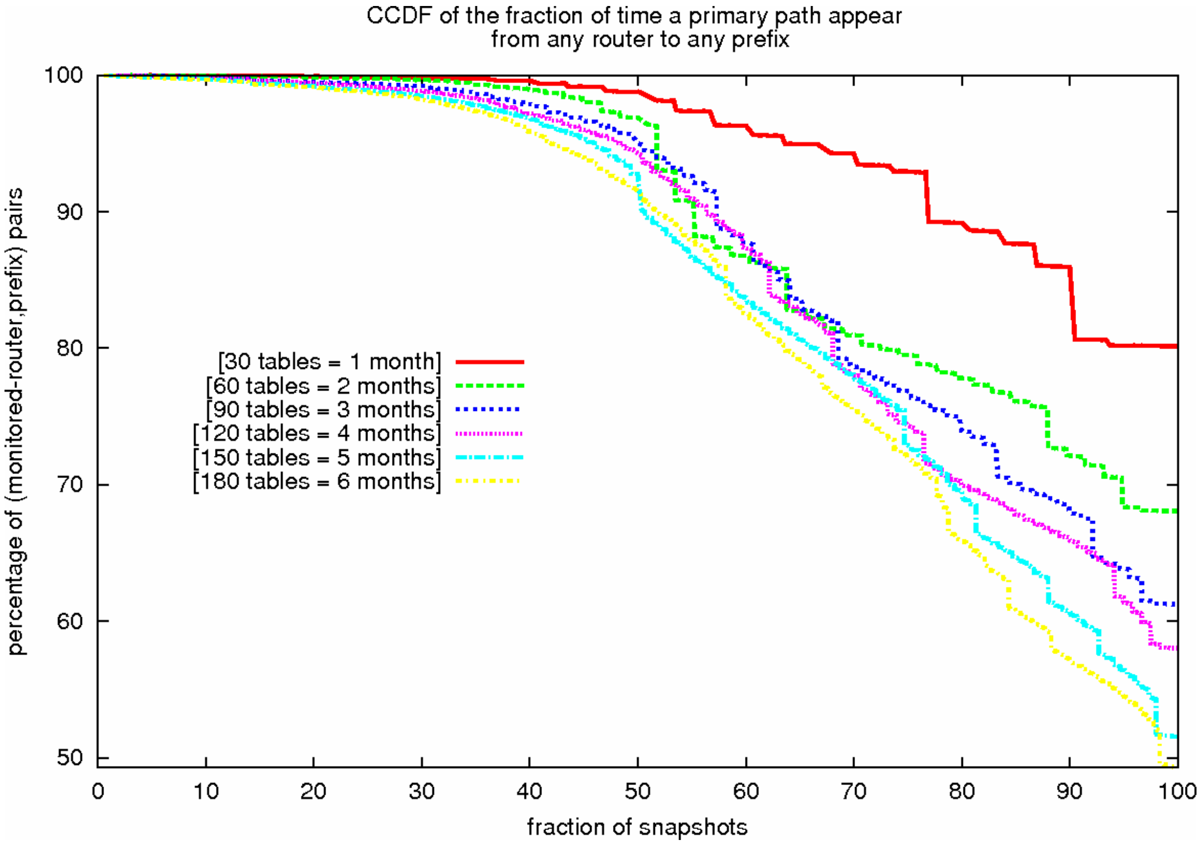
- Limitations:
  - The time is not a suitable criterion to decompose the stream of updates, it cannot address the following issues:
    - Big events cannibalizing small ones
    - Events being merged
    - An event being divided into several
  - Furthermore, such approaches have to wait for the whole consequences of events to be visible before being able to detect them

Assumptions and results founding our methodology

A preferred path for each router r to reach each prefix p

– This path, the most used for months, is called the primary path of r to p.



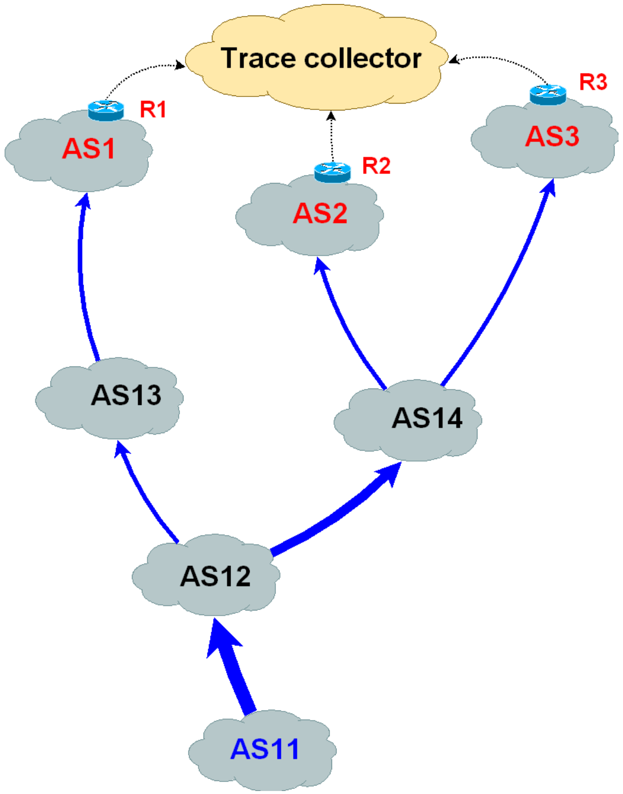


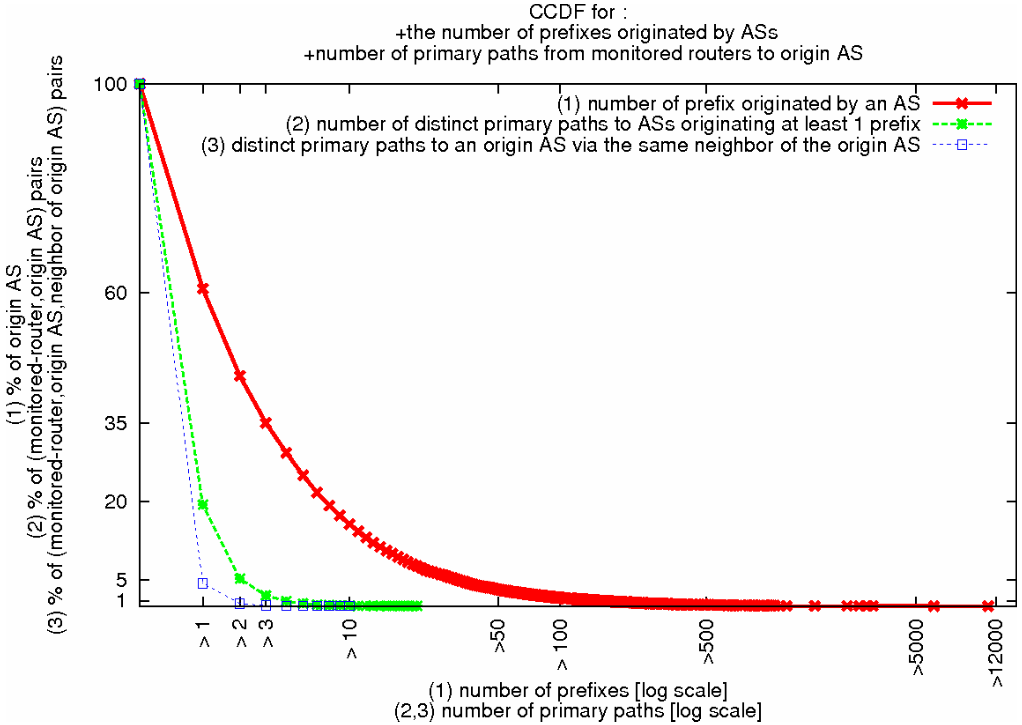
=> The events we are interested in are only temporary !

We therefore focus on detecting when a primary path becomes unavailable.

Few preferred paths to reach each AS

- A router uses generally the same primary path to reach all prefixes from a given AS
- Given a set of routers, we call their primary paths to the prefixes originated by ASx, the tree of origin AS ASx (T(ASx)).



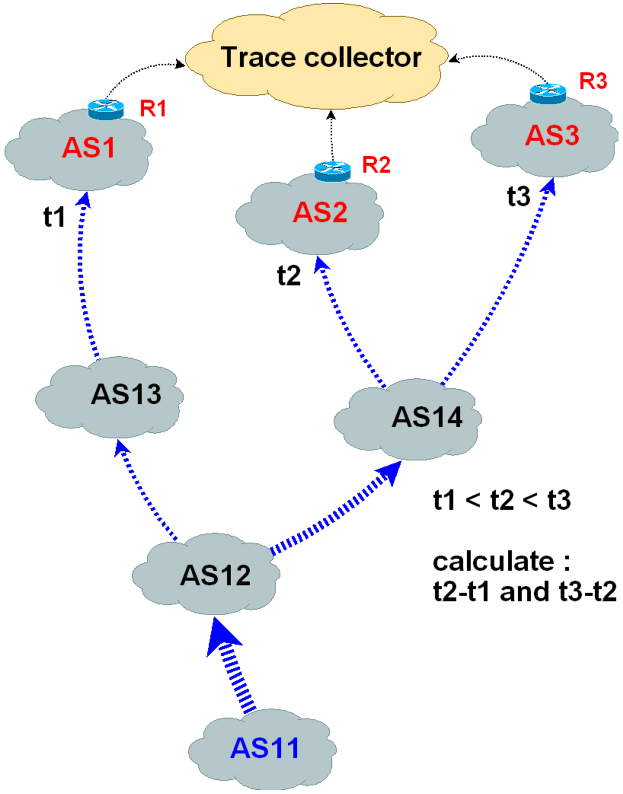


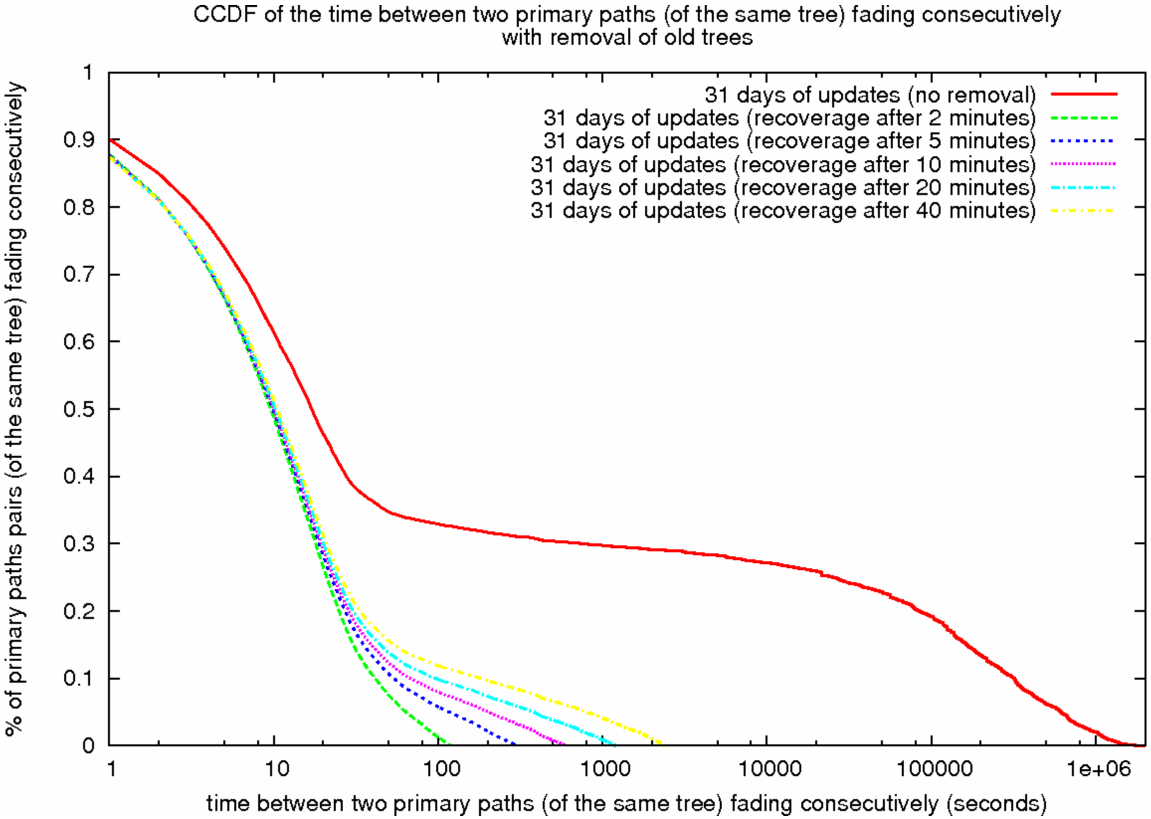
=> The interdomain routing is a routing from neighbors to neighbors rather than from prefixes to neighbors

The primary paths of a given tree are more likely impacted by the same events.

Fading trees: inter-fading time

– The updates meaning that the primary paths of a given tree are no longer available do not propagate randomly.





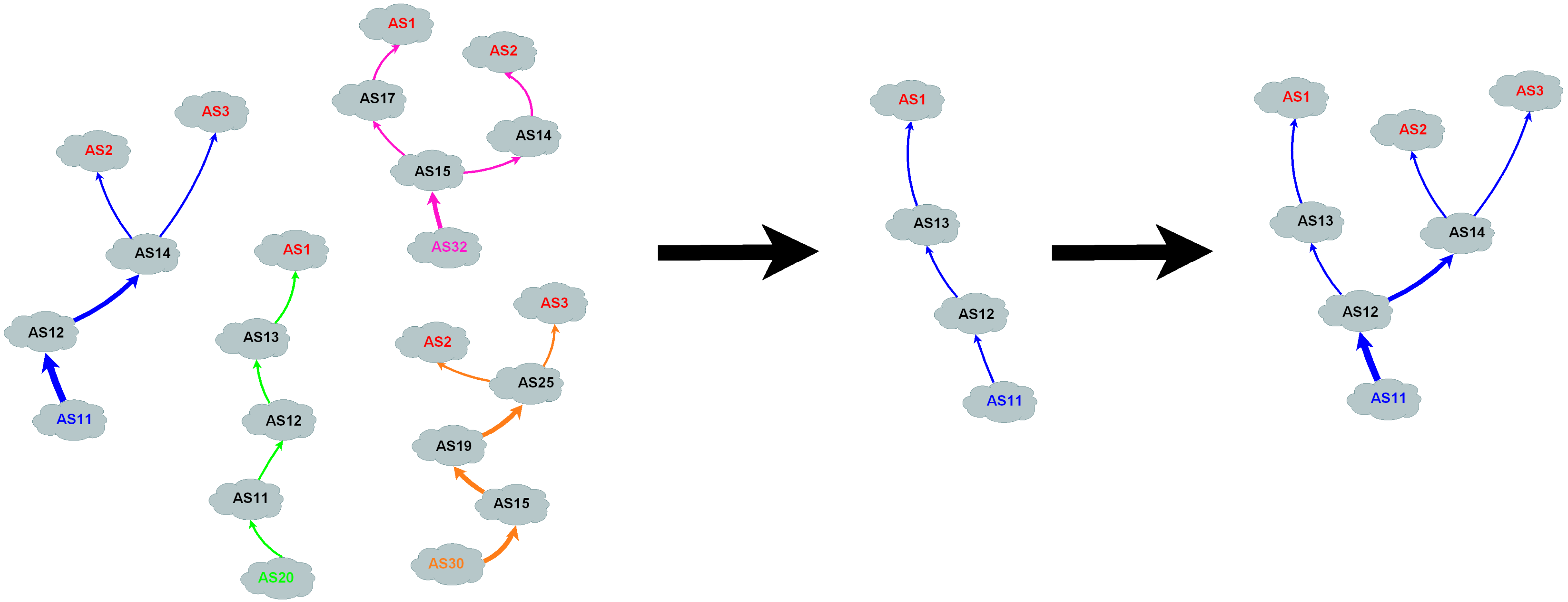
=> Conditional fading probability:

For any fading tree, if a primary path becomes unavailable at time t, then the probability to see new unavailabilities after t+2min (if you did not observe any other between t and t+2min) is very low !

Steps of the method

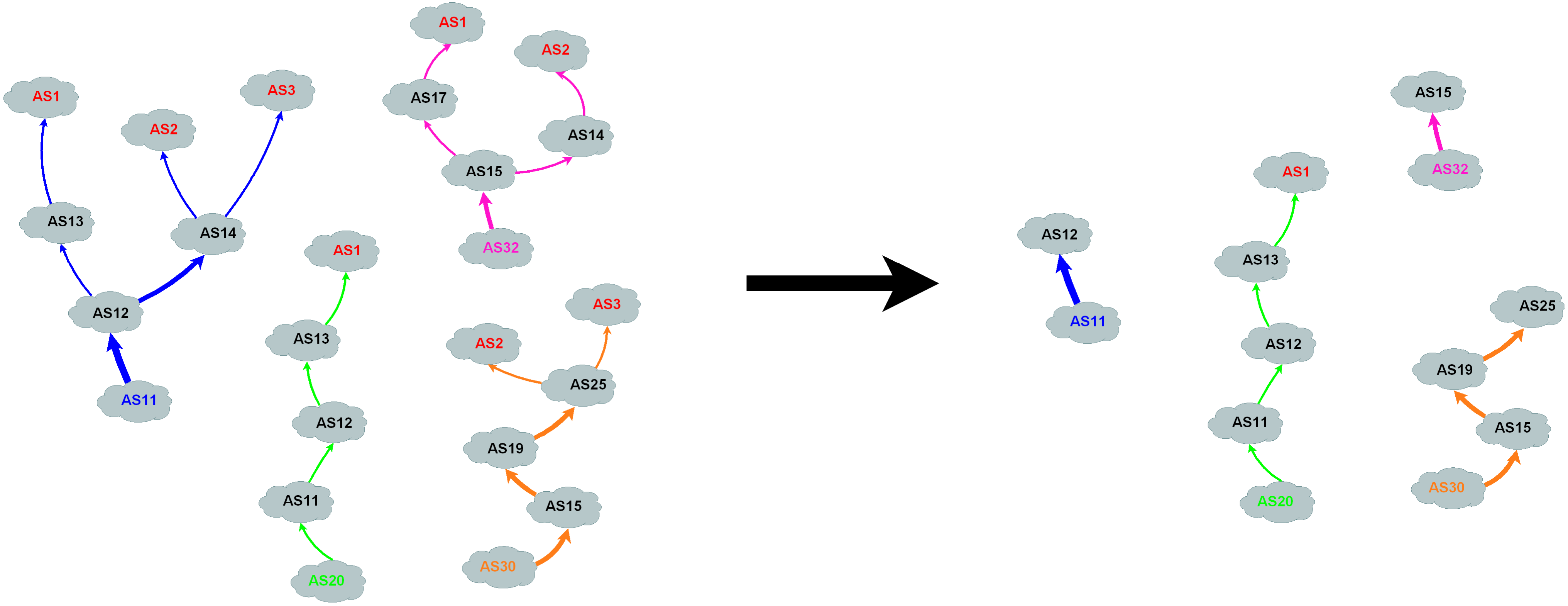
Arrival of updates and update of trees

- Given a set of fading trees (A tree is considered fading whenever at least one of its primary path is unavailable. We only represent those faded paths.)
- Whenever an update arrives meaning some primary path is no longer available, then the associated tree is updated.
- The probability to get further information in the future is then calculated for each tree based on the date of the last faded primary path.



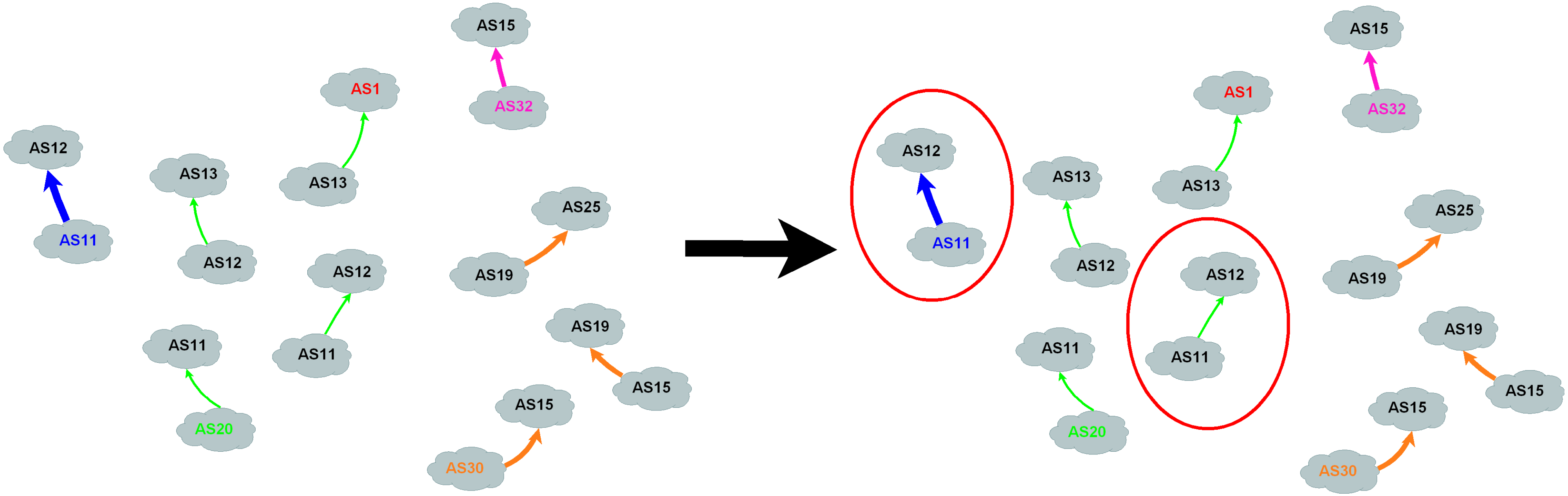
Reduction of trees

- For each tree, we only consider the common sub-path to all its faded primary paths, as the underlying event is more likely on this sub-path.



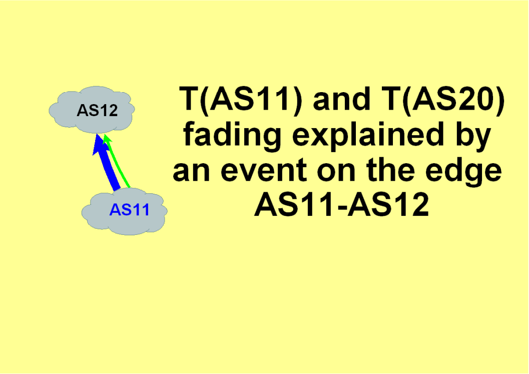
Correlation and Localization (1/2)

- For each reduced tree, the extracted sub-path is decomposed into edges.
- Those edges that belong to several trees are then identified, starting by the trees that had the more faded. (Greedy Heuristic Bigger First)
- The edges which are no longer useful are removed.

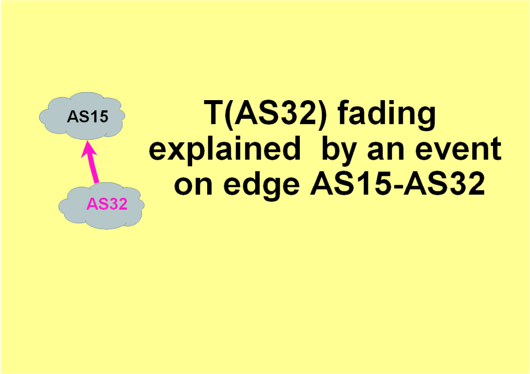


Correlation and Localization (2/2)

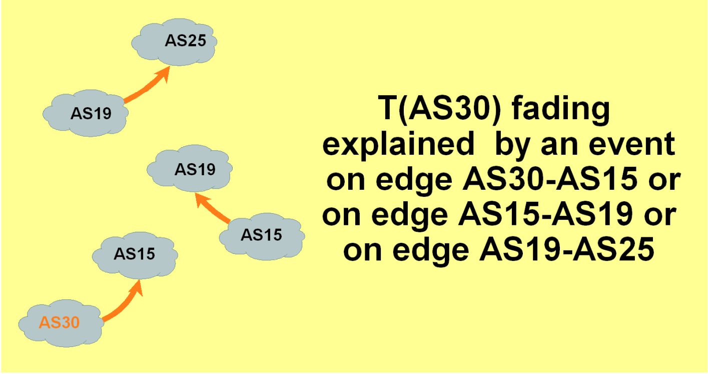
– So far we have explained the fading of the four trees by three events.



T(AS11) and T(AS20) fading explained by an event on the edge AS11-AS12



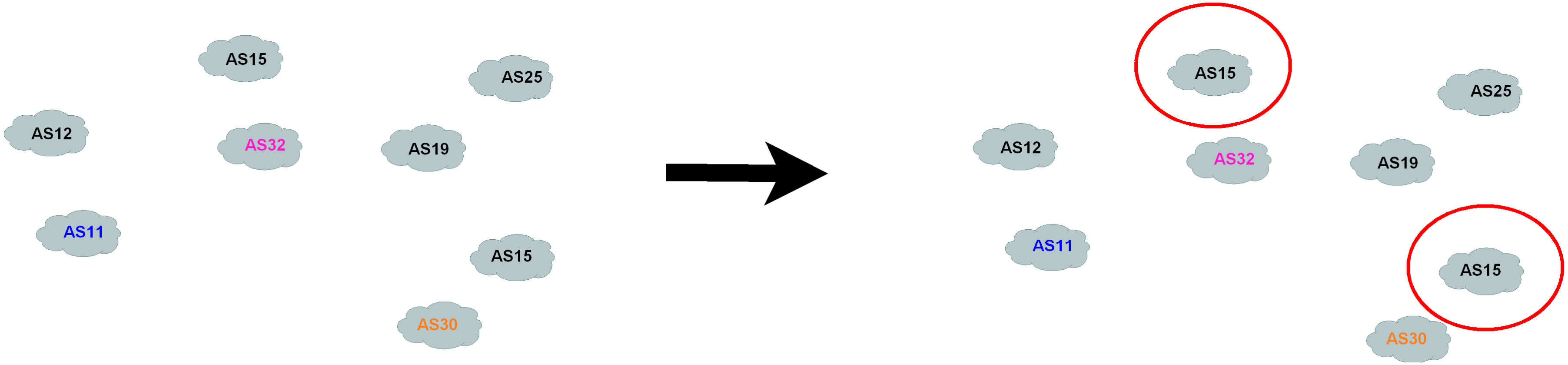
T(AS32) fading explained by an event on edge AS15-AS32



T(AS30) fading explained by an event on edge AS30-AS15 or on edge AS15-AS19 or on edge AS19-AS25

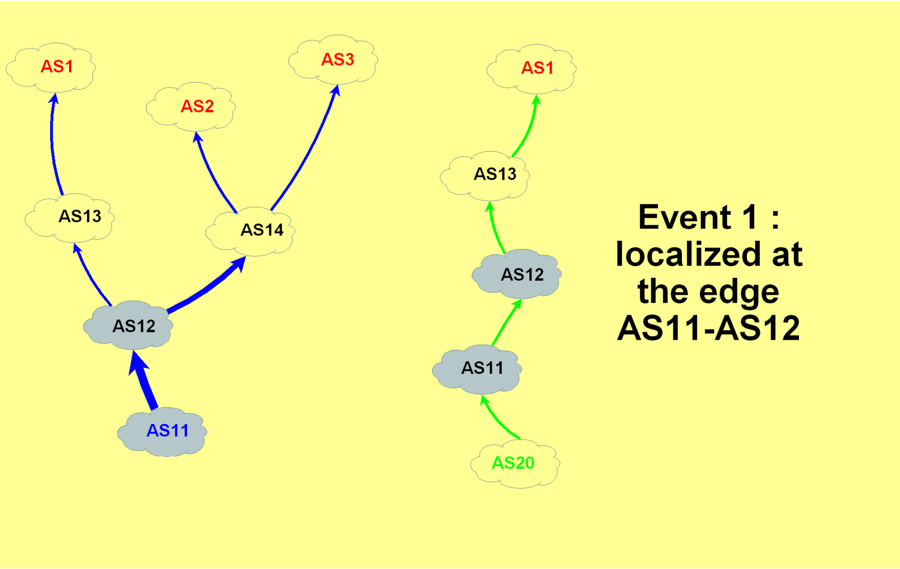
– We try to improve the result obtained applying our greedy heuristic on nodes.

– We decompose the edges obtained into nodes.

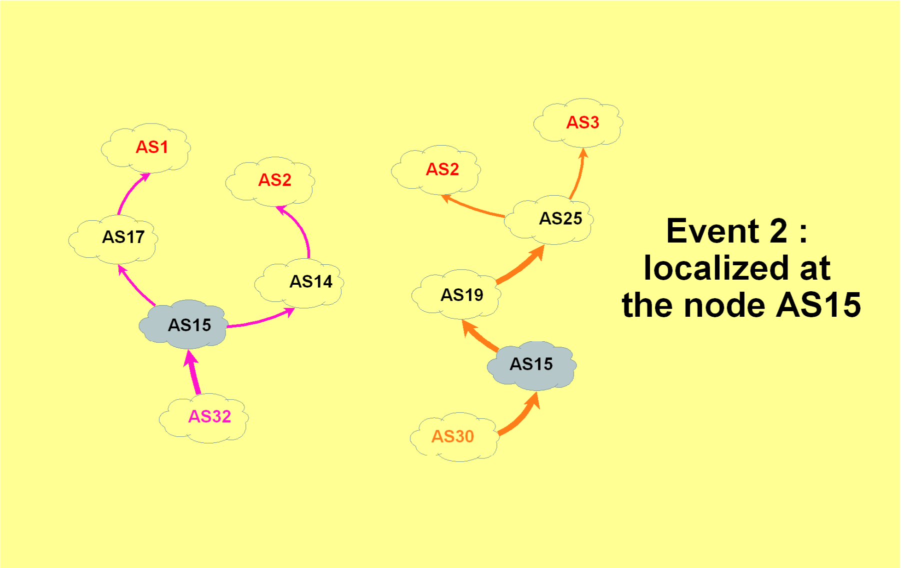


Extraction of events

- We finally obtain 2 events: both explain the fading of 2 trees, the first one is located at an edge, the second at a node.



Event 1 : localized at the edge AS11-AS12



Event 2 : localized at the node AS15

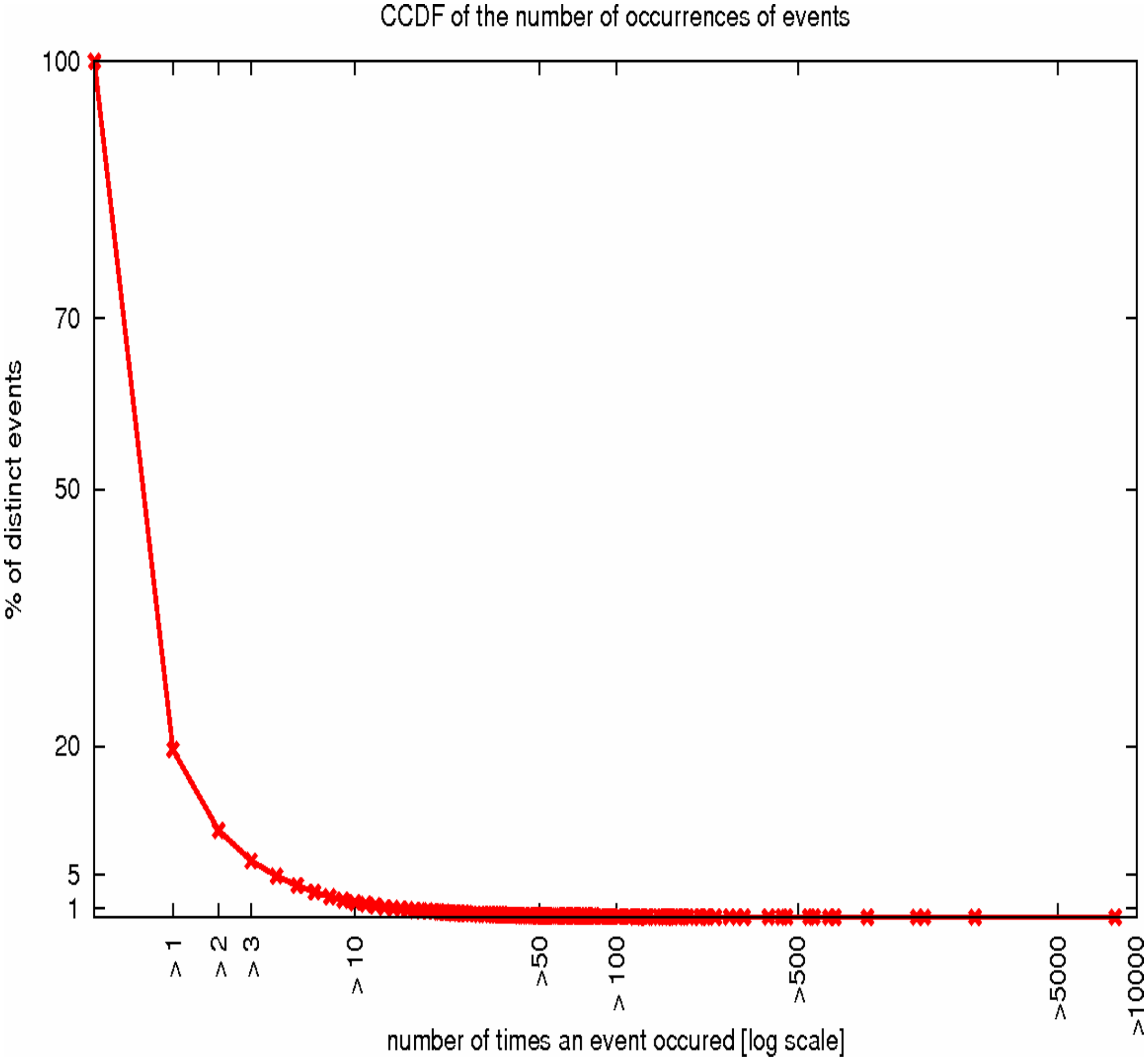
- So far we have transformed the stream of updates into a stream of events.
- An event explains the fading of some trees.
- We decide to extract an event, if and only if the conditional fading probability for each fading tree explained by this event is equal to 0.

Results and validation (1/2)

- Input and basic stats:
  - Analysis of a month of updates collected at 16 routers in 10 different ASs (Routeviews' peers at Linx)
  - 37 325 821 per-prefix announcements observed:
    - 17% (6 239 435) indicates the unavailability of a primary path.
    - 18% (6 641 341) indicates the availability of a primary path.
  - 134 674 events detected but only 64 154 distinct events

- Validation:
  - realized upon outage tickets from the AS Open Transit (AS5511)
  - Peer failures are successfully detected at the right time of their happening
  - Small recurrent events invisible so far, are also detected !

Results and validation (2/2)



- Most events happened only once (80%)
- But some events also happened hundreds or thousands of times in the month!

=> Beware of small recurrent events !

Summary of contributions

- Our method is designed to cope with the challenges of Interdomain Root Cause Analysis and to inform the user that an event has occurred as early as its effects are visible.
- The methodology breaks with previous approaches as time is not used to clusterize updates supposed to be related to the same event.
- Instead, we rely on topological considerations to correlate information until we obtain events.
- Time information is only used at the end to extract events.
- What is more, we do not use any arbitrary threshold, but rely on the contrary on empirical criteria derived from key characteristics of the interdomain routing behavior.

Ongoing job

- We are automating the tool to analyse the updates collected by routeviews’peers as soon as they are available.
- We will display the results on a public web site.
- We hope theses results will help fixing some routing problems and we will need your comments on these results !
- Finally, the more diversity we see, the more events we can detect. So, keep on or start peering with routeviews, it may benefit you !

Thanks ! Any question ?