Failure Control in Multipath Route Tracing

Darryl Veitch[1]
dveitch@unimelb.edu.au

.....

Collaborators

Brice Augustin[2] Renata Teixeira[2] Timur Friedman[2]

[1]CUBIN, Department of Electrical & Electronic Engineering, University of Melbourne, Australia

[2]UPMC Paris Universitas, and CNRS, LIP6 Laboratory, Paris, France

Traceroute, a Useful Tool

The Tool

- Aims to discover IP-level path between source and destination
- Time to Live (TTL) field in IP header targets hop-depth
- Returning TTL expiry packets carry address of last interface

Its Applications

- Diagnosing network problems
- Drawing network maps

Traceroute, a Useful Tool

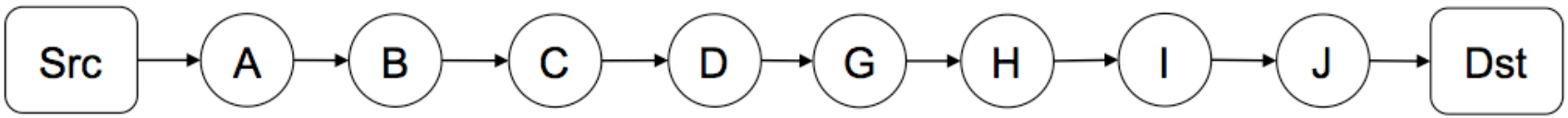
The Tool

- Aims to discover IP-level path between source and destination
- Time to Live (TTL) field in IP header targets hop-depth
- Returning TTL expiry packets carry address of last interface

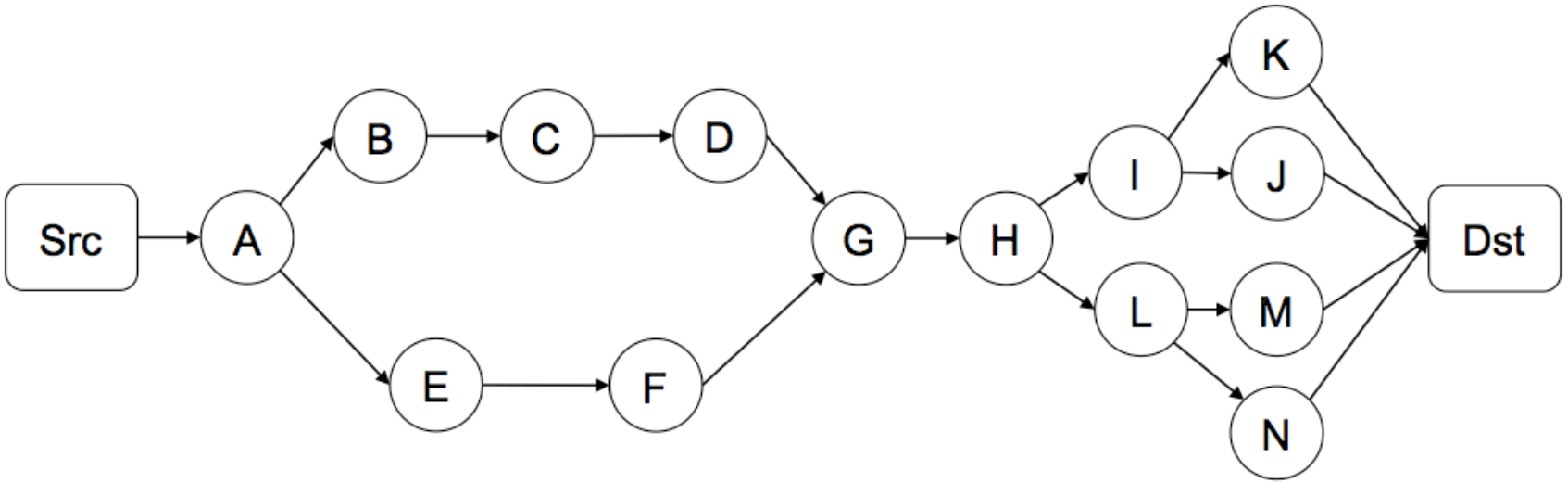
Its Applications

- Diagnosing network problems
- Drawing network maps

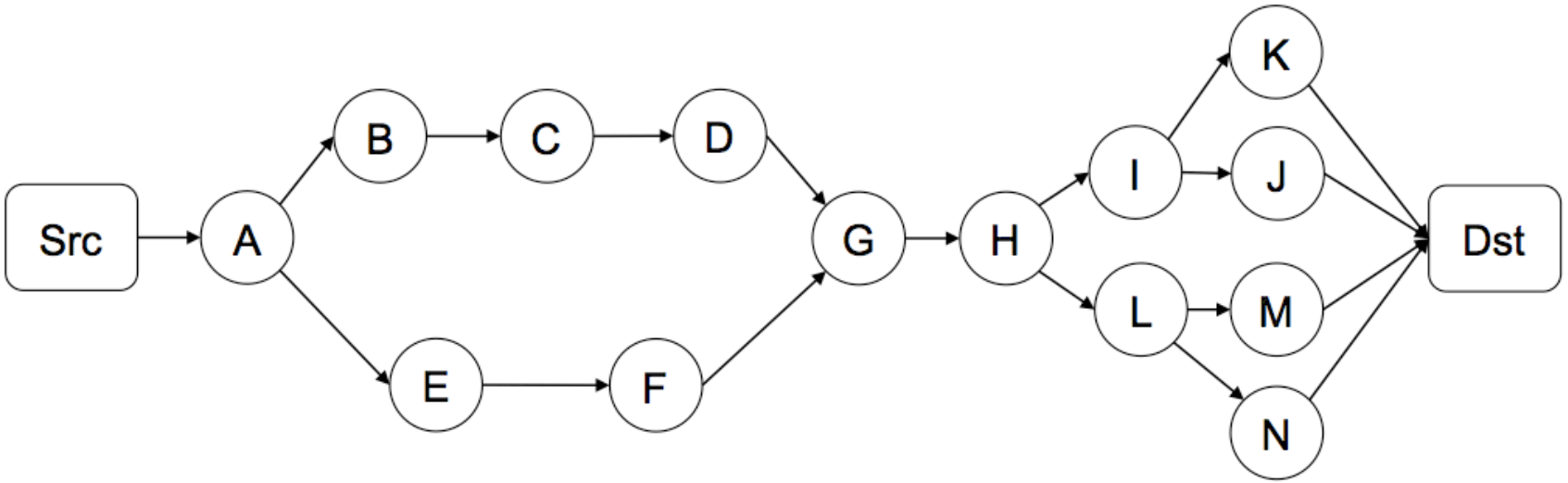
CLASSIC TRACEROUTE IGNORES LOAD BALANCING



Classic Traceroute Ignores Load Balancing



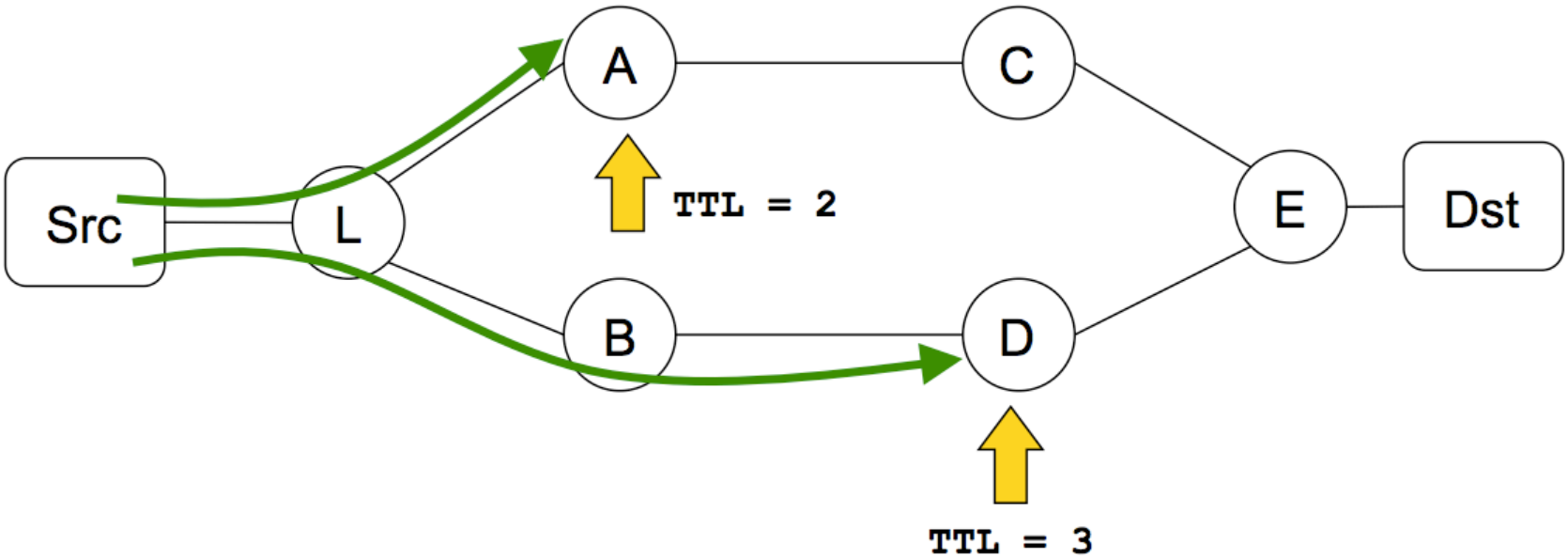
Classic Traceroute Ignores Load Balancing



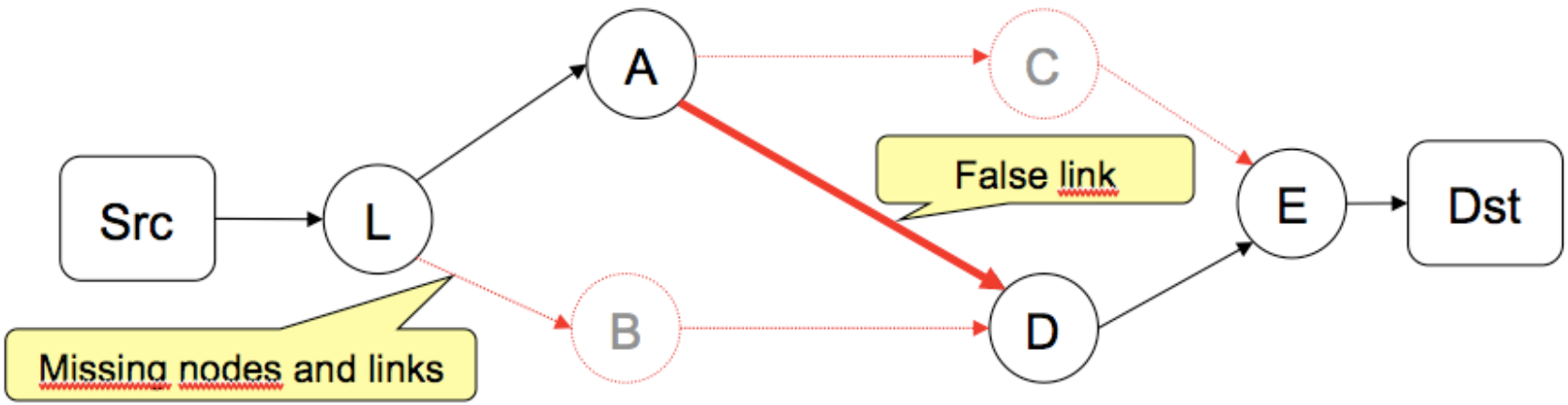
Impact on Classic Traceroute

- Paths missed
- Inferred paths erroneous

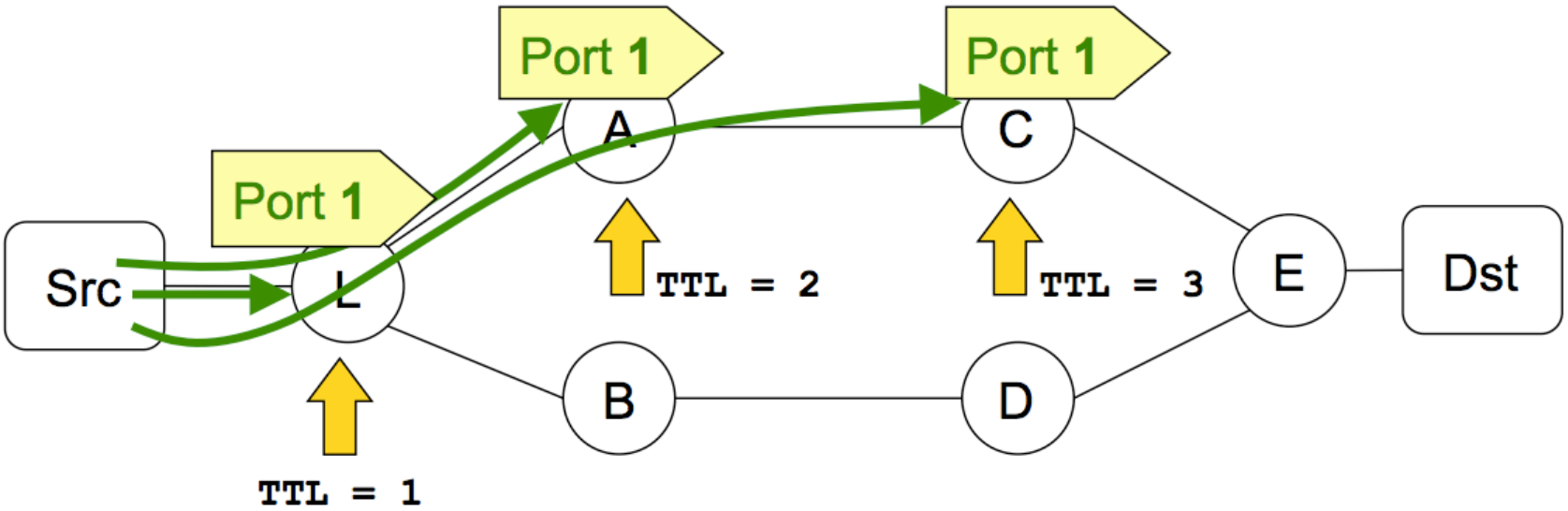
CLASSIC FAILURES



CLASSIC FAILURES



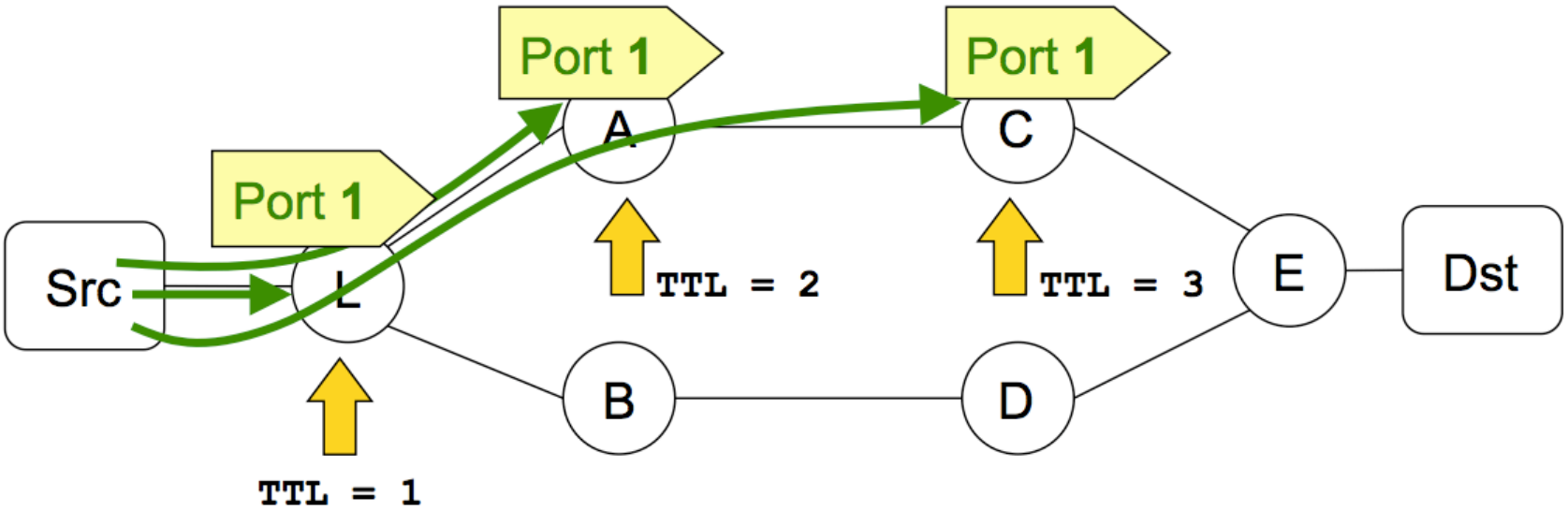
PARIS TRACEROUTE SOLUTION FOR PER-FLOW LB



EXPLICITLY CONTROL FLOW KEYS

- Ensures probes follow a fixed path assuming per-flow LB
- Failure to do so reveals presence of per-packet LB

PARIS TRACEROUTE SOLUTION FOR PER-FLOW LB



EXPLICITLY CONTROL FLOW KEYS

- Ensures probes follow a fixed path assuming per-flow LB
- Failure to do so reveals presence of per-packet LB

Paris Traceroute Development

Our Prior Work

- **IMC'06**:
  – Revealing flaws in Classic Traceroute
  – Introduction of flow key fixing idea for single path under per-flow LB
- **E2EMon'07**: Introduction of
  – Stochastic probing algorithm Multipath detection algorithm (MDA)
  – Per-node significance levels
- **IMC'07**: Use of MDA to study LB prevalence in Internet

This Paper

- Multipath failure probabilities, calculation and bounds
- Extension from node to path level statistical guarantees
- Exact significance levels for multipath discovery

Paris Traceroute Development

Our Prior Work

- **IMC'06**:
  – Revealing flaws in Classic Traceroute
  – Introduction of flow key fixing idea for single path under per-flow LB
- **E2EMon'07**: Introduction of
  – Stochastic probing algorithm Multipath detection algorithm (MDA)
  – Per-node significance levels
- **IMC'07**: Use of MDA to study LB prevalence in Internet

This Paper

- Multipath failure probabilities, calculation and bounds
- Extension from node to path level statistical guarantees
- Exact significance levels for multipath discovery

Multipath Route Discovery

Adaptive probing strategy: at each hop

- Controlled exploration of multipath by varying flow identifier of probes
- Send enough probes to enumerate all interfaces with high confidence (retransmit if necessary)
- Classify load balancers: per-flow or per-packet

Understanding Failure

Failure

- Means not finding all links
- Operates at node and graph level
- Implies topology known! but we want to *discover* multipath...
- Importance of failure:
  - gives control for fixed (common) topologies
  - provides the foundation for everything

Assumptions for Calculation

- Node level: probe returns interface IID uniformly out of $\{1, 2, \ldots K\}$
- Graph level: all nodes independent

Understanding Failure

Failure

- Means not finding all links
- Operates at node and graph level
- Implies topology known! but we want to *discover* multipath...
- Importance of failure:
  – gives control for fixed (common) topologies
  – provides the foundation for everything

Assumptions for Calculation

- Node level: probe returns interface IID uniformly out of $\{1, 2, \ldots K\}$
- Graph level: all nodes independent

Understanding Failure

Failure

- Means not finding all links
- Operates at node and graph level
- Implies topology known! but we want to *discover* multipath...
- Importance of failure:
  – gives control for fixed (common) topologies
  – provides the foundation for everything

Assumptions for Calculation

- Node level: probe returns interface IID uniformly out of $\{1, 2, \ldots K\}$
- Graph level: all nodes independent

Failure Probabilities

Node Failure Probability β_K

Define stopping points $\{n_k\}$ for each of $k \in \{1, 2, \ldots K\}$

Corresponding *stopping probability* t_K^k at level k $\quad (\sum_{k=1}^{K} t_K^k = 1)$

$$\begin{aligned} \beta_1 &= 0 \\ \beta_K &= \sum_{k=1}^{K-1} t_K^k, \quad K > 1 \end{aligned}$$

Failure Probabilities

Node Failure Probability β_K

Define stopping points $\{n_k\}$ for each of $k \in \{1, 2, \ldots K\}$

Corresponding *stopping probability* t_K^k at level k $\quad (\sum_{k=1}^{K} t_K^k = 1)$

$$\begin{aligned} \beta_1 &= 0 \\ \beta_K &= \sum_{k=1}^{K-1} t_K^k, \quad K > 1 \end{aligned}$$

Failure Probabilities

Node Failure Probability β_K

Define stopping points $\{n_k\}$ for each of $k \in \{1, 2, \ldots K\}$

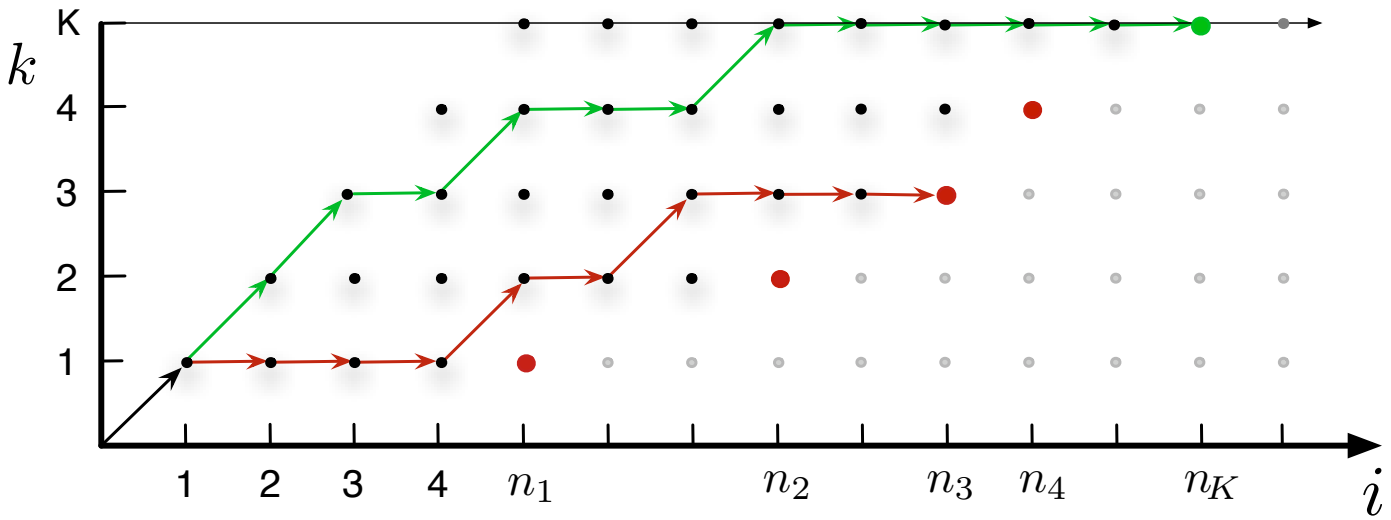
Corresponding *stopping probability* t_K^k at level k $\quad (\sum_{k=1}^{K} t_K^k = 1)$

$$\begin{aligned} \beta_1 &= 0 \\ \beta_K &= \sum_{k=1}^{K-1} t_K^k, \quad K > 1 \end{aligned}$$

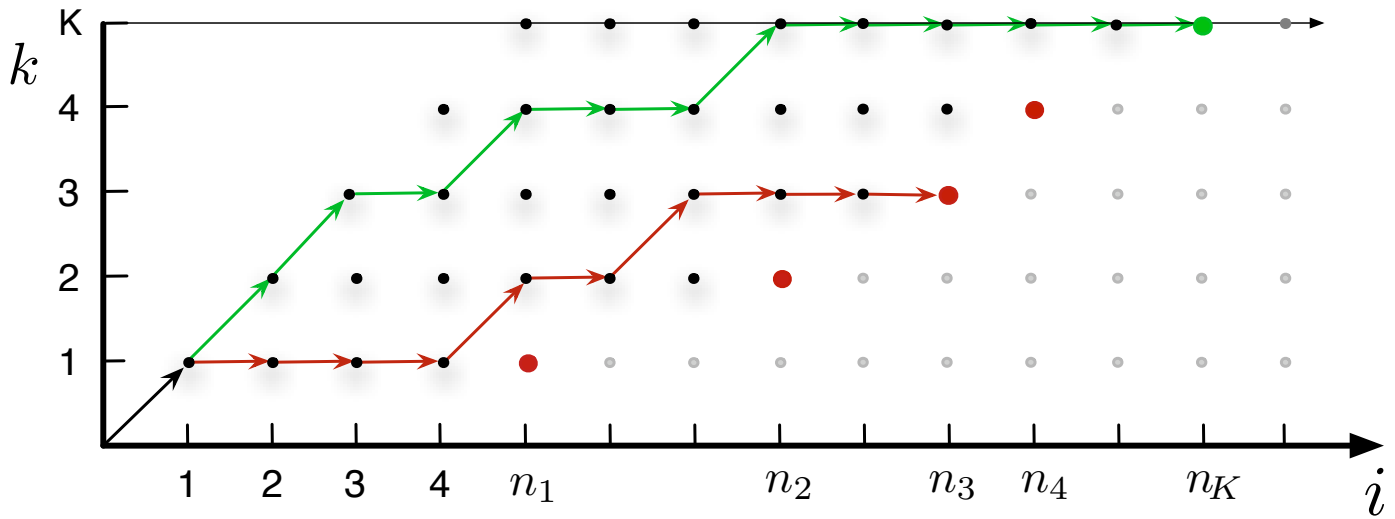
Graph Failure Probability β_{all}

$$\beta_{\text{all}} = 1 - \prod_i (1 - \beta_{K_i})$$

Node State Space and Trajectories

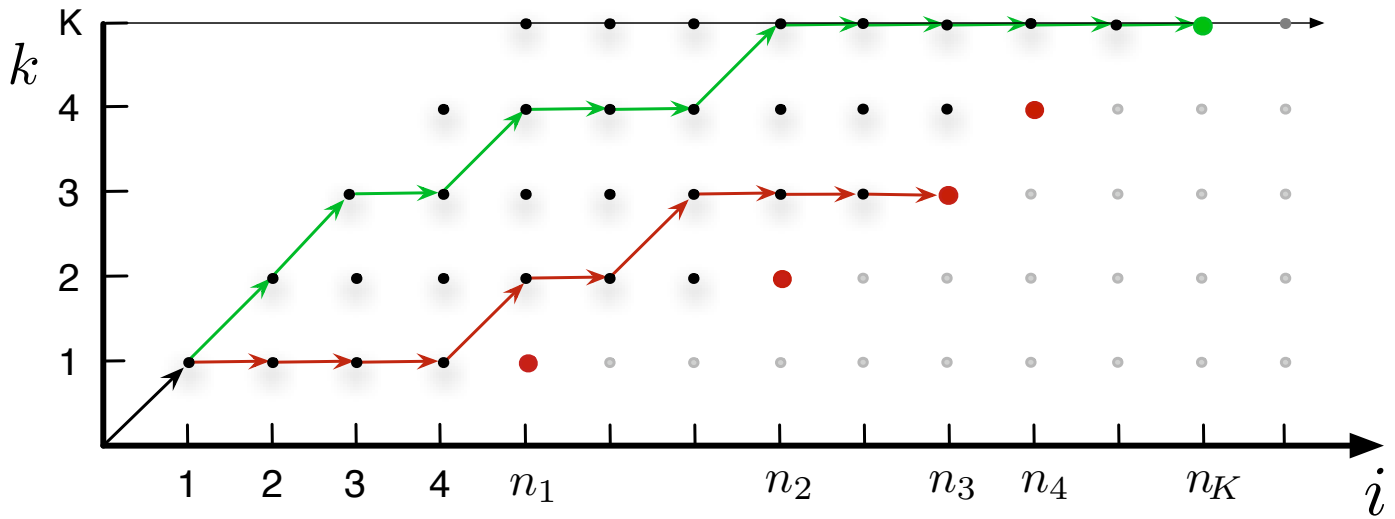


Node State Space and Trajectories



- Transition probabilities after new probe:
  - Find new: $q_k = (K-k)/K$
  - Nothing new: $p_k = 1 - q_k = k/K$
- To reach $(n_1, 1)$: $t_K^1 = p_1^{n_1-1}$; For $(n_2, 2)$: $t_K^2 = \sum_{j_1=0}^{n_1-2} p_1^{j_1} q_1 p_2^{n_2-2-j_1}$

Node State Space and Trajectories



- Transition probabilities after new probe:
  - Find new: $q_k = (K-k)/K$
  - Nothing new: $p_k = 1 - q_k = k/K$
- To reach $(n_1, 1)$: $t_K^1 = p_1^{n_1-1}$; For $(n_2, 2)$: $t_K^2 = \sum_{j_1=0}^{n_1-2} p_1^{j_1} q_1 p_2^{n_2-2-j_1}$

Conditional Failure Probabilities

Stopping probabilities t_K^k are unconditional.
For adaptive algorithm, want conditional ones *given j interfaces already seen*.

$$\begin{aligned} t_{K,j}^k &= \Pr(\text{stop at level } k \mid \text{didn't stop at levels below } j) \\ &= \frac{\Pr(\text{stop at level } k)}{\Pr(\text{didn't stop at levels below } j)} = \frac{t_K^k}{\sum_{l=j}^K t_K^l} \end{aligned}$$

for $k \geqslant j$ and zero otherwise.
Corresponding conditional failure probability, $1 \leqslant j \leqslant K$, is

$$\begin{aligned} \beta_{K,K} &= 0 \\ \beta_{K,j} &= \sum_{k=1}^{K-1} t_{K,j}^k = \sum_{k=j}^{K-1} t_{K,j}^k = \frac{\sum_{k=j}^{K-1} t_K^k}{\sum_{l=j}^K t_K^l}, \quad j < K, \end{aligned}$$

and naturally $\beta_{K,1} = \beta_K$.

Conditional Failure Probabilities

Stopping probabilities t_K^k are unconditional.
For adaptive algorithm, want conditional ones *given j interfaces already seen*.

$$\begin{aligned} t_{K,j}^k &= \Pr(\text{stop at level } k \mid \text{didn't stop at levels below } j) \\ &= \frac{\Pr(\text{stop at level } k)}{\Pr(\text{didn't stop at levels below } j)} = \frac{t_K^k}{\sum_{l=j}^K t_K^l} \end{aligned}$$

for $k \geqslant j$ and zero otherwise.

Corresponding conditional failure probability, $1 \leqslant j \leqslant K$, is

$$\begin{aligned} \beta_{K,K} &= 0 \\ \beta_{K,j} &= \sum_{k=1}^{K-1} t_{K,j}^k = \sum_{k=j}^{K-1} t_{K,j}^k = \frac{\sum_{k=j}^{K-1} t_K^k}{\sum_{l=j}^K t_K^l}, \quad j < K, \end{aligned}$$

and naturally $\beta_{K,1} = \beta_K$.

Conditional Failure Probabilities

Stopping probabilities t_K^k are unconditional.
For adaptive algorithm, want conditional ones *given j interfaces already seen*.

$$\begin{aligned} t_{K,j}^k &= \Pr(\text{stop at level } k \mid \text{didn't stop at levels below } j) \\ &= \frac{\Pr(\text{stop at level } k)}{\Pr(\text{didn't stop at levels below } j)} = \frac{t_K^k}{\sum_{l=j}^{K} t_K^l} \end{aligned}$$

for $k \geqslant j$ and zero otherwise.
Corresponding conditional failure probability, $1 \leqslant j \leqslant K$, is

$$\begin{aligned} \beta_{K,K} &= 0 \\ \beta_{K,j} &= \sum_{k=1}^{K-1} t_{K,j}^k = \sum_{k=j}^{K-1} t_{K,j}^k = \frac{\sum_{k=j}^{K-1} t_K^k}{\sum_{l=j}^{K} t_K^l}, \quad j < K, \end{aligned}$$

and naturally $\beta_{K,1} = \beta_K$.

Hypothesis Testing Framework

In practice, K unknown. Role of algorithm is to give up search reluctantly:

Control probability α_k of stopping at state (n_k, k), given have k interfaces already, under null hypothesis that in fact are $K' > k$ of them.

Null-hyp is composite, but since $p_{k,K'} = k/K'$, sig-level set by $K' = k+1$.

Simple Null Hypothesis of $K' = k+1$

- Stopping at (n_k, k) and failing to find new successors is a type-I error
- Stopping accepts alternate hypothesis that $K' = k$
- Test has optimal power! (if $k = K' = K$, impossible to find more)
- Significance level is

$$\alpha_k = t^k_{k+1,k} = \frac{t^k_{k+1}}{t^k_{k+1} + t^{k+1}_{k+1}}, \quad 1 \leqslant k \leqslant K.$$

Hypothesis Testing Framework

In practice, K unknown. Role of algorithm is to give up search reluctantly:

Control probability α_k of stopping at state (n_k, k), given have k interfaces already, under null hypothesis that in fact are $K' > k$ of them.

Null-hyp is composite, but since $p_{k,K'} = k/K'$, sig-level set by $K' = k+1$.

Simple Null Hypothesis of $K' = k+1$

- Stopping at (n_k, k) and failing to find new successors is a type-I error
- Stopping accepts alternate hypothesis that $K' = k$
- Test has optimal power! (if $k = K' = K$, impossible to find more)
- Significance level is

$$\alpha_k = t^k_{k+1,k} = \frac{t^k_{k+1}}{t^k_{k+1} + t^{k+1}_{k+1}}, \quad 1 \leqslant k \leqslant K.$$

Connecting Significance and Failure

Upon terminating a node-level discovery, a significance level will be reported.

How does it relate to the true (conditional) probability of error?

Relationship not Simple!

- If $k = K$ found, then $\alpha_K > \beta_{K,K} = 0$, so algorithm is conservative
- If $k = K - 1$, algo uses $K' = K$, and $\alpha_{K-1} = \beta^{K'}_{K,K-1} = \beta_{K,K-1}$, ideal!
- If $k < K - 1$, can go either way, e.g. with $K = 3$ and $k = 1$:
  - If $(n_1, n_2, n_3) = (2, 30, 30)$ then $\alpha_1 > \beta_{3,1}$ (pessimistic, conservative)
  - If $(n_1, n_2, n_3) = (2, 5, 4)$ ... find $\alpha_1 < \beta_{3,1}$ (optimistic, not conservative)

Connecting Significance and Failure

Upon terminating a node-level discovery, a significance level will be reported.

How does it relate to the true (conditional) probability of error?

Relationship not Simple!

- If $k = K$ found, then $\alpha_K > \beta_{K,K} = 0$, so algorithm is conservative
- If $k = K - 1$, algo uses $K' = K$, and $\alpha_{K-1} = t^{K-1}_{K,K-1} = \beta_{K,K-1}$, ideal!
- If $k < K - 1$, can go either way, e.g. with $K = 3$ and $k = 1$:
  - If $(n_1, n_2, n_3) = (2, 30, 30)$ then $\alpha_1 > \beta_{3,1}$ (pessimistic, conservative)
  - If $(n_1, n_2, n_3) = (2, 3, 4)$, find $\alpha_1 < \beta_{3,1}$ (optimistic, misleading!)

Connecting Significance and Failure

Upon terminating a node-level discovery, a significance level will be reported.

How does it relate to the true (conditional) probability of error?

Relationship not Simple!

- If $k = K$ found, then $\alpha_K > \beta_{K,K} = 0$, so algorithm is conservative
- If $k = K - 1$, algo uses $K' = K$, and $\alpha_{K-1} = t^{K-1}_{K,K-1} = \beta_{K,K-1}$, ideal!
- If $k < K - 1$, can go either way, e.g. with $K = 3$ and $k = 1$:
  - If $(n_1, n_2, n_3) = (2, 30, 30)$ then $\alpha_1 > \beta_{3,1}$ (pessimistic, conservative)
  - If $(n_1, n_2, n_3) = (2, 3, 4)$, find $\alpha_1 < \beta_{3,1}$ (optimistic, misleading!)

Connecting Significance and Failure

Upon terminating a node-level discovery, a significance level will be reported.

How does it relate to the true (conditional) probability of error?

Relationship not Simple!

- If $k = K$ found, then $\alpha_K > \beta_{K,K} = 0$, so algorithm is conservative
- If $k = K - 1$, algo uses $K' = K$, and $\alpha_{K-1} = t_{K,K-1}^{K-1} = \beta_{K,K-1}$, ideal!
- If $k < K - 1$, can go either way, e.g. with $K = 3$ and $k = 1$:
  - If $(n_1, n_2, n_3) = (2, 30, 30)$ then $\alpha_1 > \beta_{3,1}$ (pessimistic, conservative)
  - If $(n_1, n_2, n_3) = (2, 3, 4)$, find $\alpha_1 < \beta_{3,1}$ (optimistic, misleading!)

Connecting Significance and Failure

Upon terminating a node-level discovery, a significance level will be reported.

How does it relate to the true (conditional) probability of error?

Relationship not Simple!

- If $k = K$ found, then $\alpha_K > \beta_{K,K} = 0$, so algorithm is conservative
- If $k = K - 1$, algo uses $K' = K$, and $\alpha_{K-1} = t_{K,K-1}^{K-1} = \beta_{K,K-1}$, ideal!
- If $k < K - 1$, can go either way, e.g. with $K = 3$ and $k = 1$:
  - If $(n_1, n_2, n_3) = (2, 30, 30)$ then $\alpha_1 > \beta_{3,1}$ (pessimistic, conservative)
  - If $(n_1, n_2, n_3) = (2, 3, 4)$, find $\alpha_1 < \beta_{3,1}$ (optimistic, misleading!)

So can we know anything for sure in practice?

FAILURE BOUNDS

Problem: can we design the $\{n_k\}$ to ensure that $\alpha_k \leq \beta_{K,k}$ for all possible K?

Failure Bounds

Problem: can we design the $\{n_k\}$ to ensure that $\alpha_k \leq \beta_{K,k}$ for all possible K?

Yes !

FAILURE BOUNDS

Problem: can we design the $\{n_k\}$ to ensure that $\alpha_k \leq \beta_{K,k}$ for all possible K?

Solution:

Step 1: Design the $\{\alpha_k\}$. Easy to show that $\alpha_k \geqslant t^k_{K,k} \geqslant t^k_K$, $\Longrightarrow$

$$\beta_{K,k} \leqslant \beta_{K,1} = \sum_{l=1}^{K-1} t^l_K \leqslant \sum_{l=1}^{K-1} \alpha_l \ .$$

Note if for example $\alpha_k = \alpha$ for all k, not useful! If set $\alpha_k = \alpha_1 r^{k-1}$:

$$\beta_{K,k} \leqslant \sum_{l=1}^{K-1} \alpha_l < \frac{\alpha_1}{1-r} \ .$$

To achieve failure target β^*, select a convenient α_1, then set $r = 1 - \frac{\alpha_1}{\beta^*}$.

Step 2: From $\{\alpha_k\}$, the corresponding $\{n_k\}$ can be calculated recursively.

Result: $\beta_{K,k} \leq \beta^*$ regardless of the unknown K !

FAILURE BOUNDS

Problem: can we design the $\{n_k\}$ to ensure that $\alpha_k \leq \beta_{K,k}$ for all possible K?

Solution:

Step 1: Design the $\{\alpha_k\}$. Easy to show that $\alpha_k \geqslant t^k_{K,k} \geqslant t^k_K$, $\Longrightarrow$

$$\beta_{K,k} \leqslant \beta_{K,1} = \sum_{l=1}^{K-1} t^l_K \leqslant \sum_{l=1}^{K-1} \alpha_l \ .$$

Note if for example $\alpha_k = \alpha$ for all k, not useful! If set $\alpha_k = \alpha_1 r^{k-1}$:

$$\beta_{K,k} \leqslant \sum_{l=1}^{K-1} \alpha_l < \frac{\alpha_1}{1-r} \ .$$

To achieve failure target β^*, select a convenient α_1, then set $r = 1 - \frac{\alpha_1}{\beta^*}$.

Step 2: From $\{\alpha_k\}$, the corresponding $\{n_k\}$ can be calculated recursively.

Result: $\beta_{K,k} \leq \beta^*$ regardless of the unknown K !

Example with $\beta^* = 0.05$

k	1	2	3	4	5	6	7	8	9
n_k	9	17	24	33	42	51	60	70	81
k	10	11	12	13	14	15	16	$\cdots$	
n_k	91	102	113	125	136	148	161	$\cdots$	

Two Kinds of Statistical Guarantees

Node level guarantees defined by $\{n_k\}$. Can be set in two ways:

- Via failure probability bound: β^*
- Via chosen significance levels: $\alpha_k = \alpha_{\text{node}}$

Graph level guarantees must also bound max size of multipath:

$$\beta^*_{\text{all}} = 1 - \prod_{i}^{\max}(1 - \beta^*_{k_i})$$

Ground Truth

True Topology Unavailable

- Network operators don't like sharing
- Network operators don't know (atypical behaviours difficult to pin down)
- Routing is constantly changing..

Our Surrogate Ground Truth

- Use MDA with very high β^* to discovery 'everything we can'
- Use same data set to evaluate smaller β^*
  – interfaces found guaranteed to be in ground truth set
  – avoids non-stationarity issue
- Provides well defined assessment of algorithm

Ground Truth

True Topology Unavailable

- Network operators don't like sharing
- Network operators don't know (atypical behaviours difficult to pin down)
- Routing is constantly changing..

Our Surrogate Ground Truth

- Use MDA with very high β^* to discovery 'everything we can'
- Use same data set to evaluate smaller β^*
  – interfaces found guaranteed to be in ground truth set
  – avoids non-stationarity issue
- Provides well defined assessment of algorithm

Data

Probes

- UDP Probes
- TTL = 36
- Inter-Probe interval: 50ms
- Retransmissions:
  – up to 3 attempts
  – timeout 2 seconds

Routes

- Source: UPMC Paris Universitas
  – 44% of routes traverse a LB (prior work gives 23% – 80%)
- Destinations: 5000 randomly selected responsive to pings
- Show 1 of 4 rounds taken over 3 weeks

Data

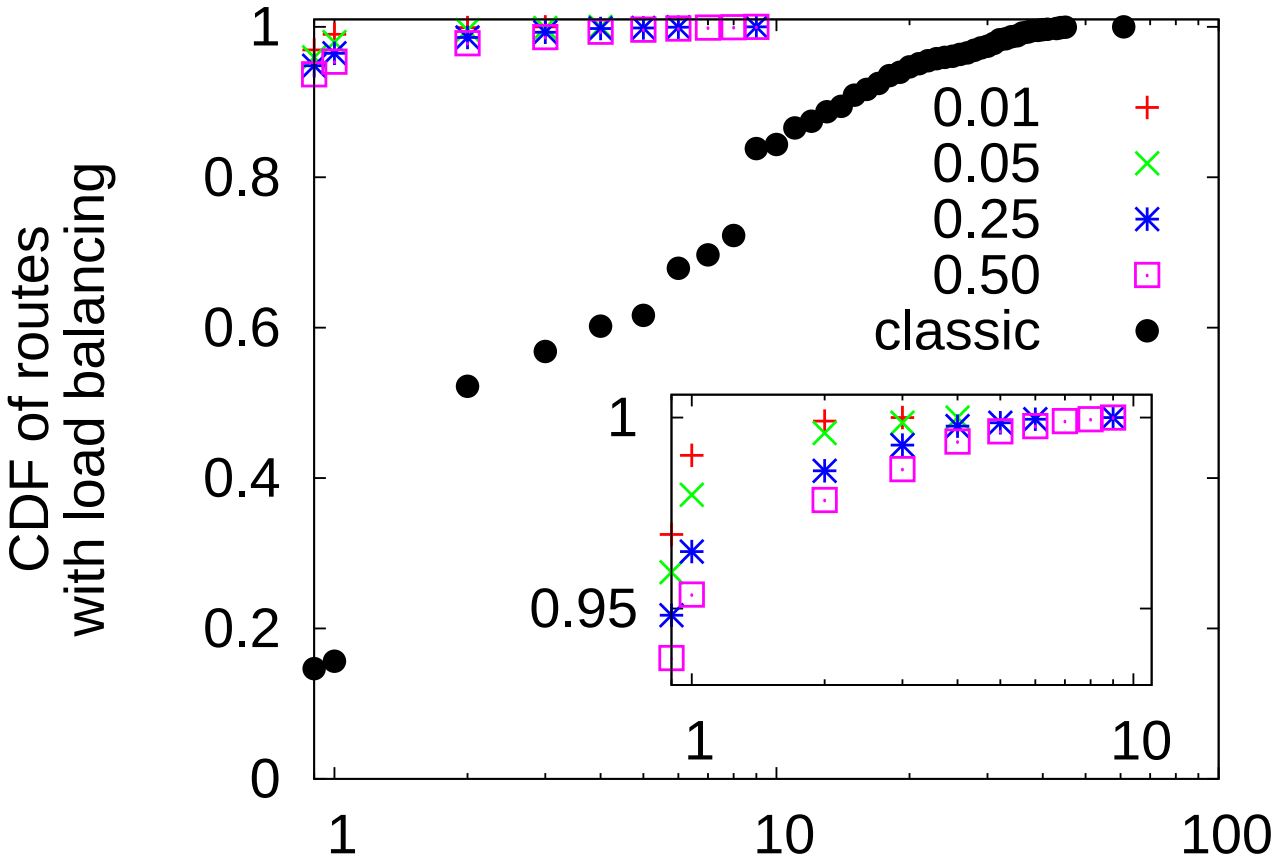
Probes

- UDP Probes
- TTL = 36
- Inter-Probe interval: 50ms
- Retransmissions:
  – up to 3 attempts
  – timeout 2 seconds

Routes

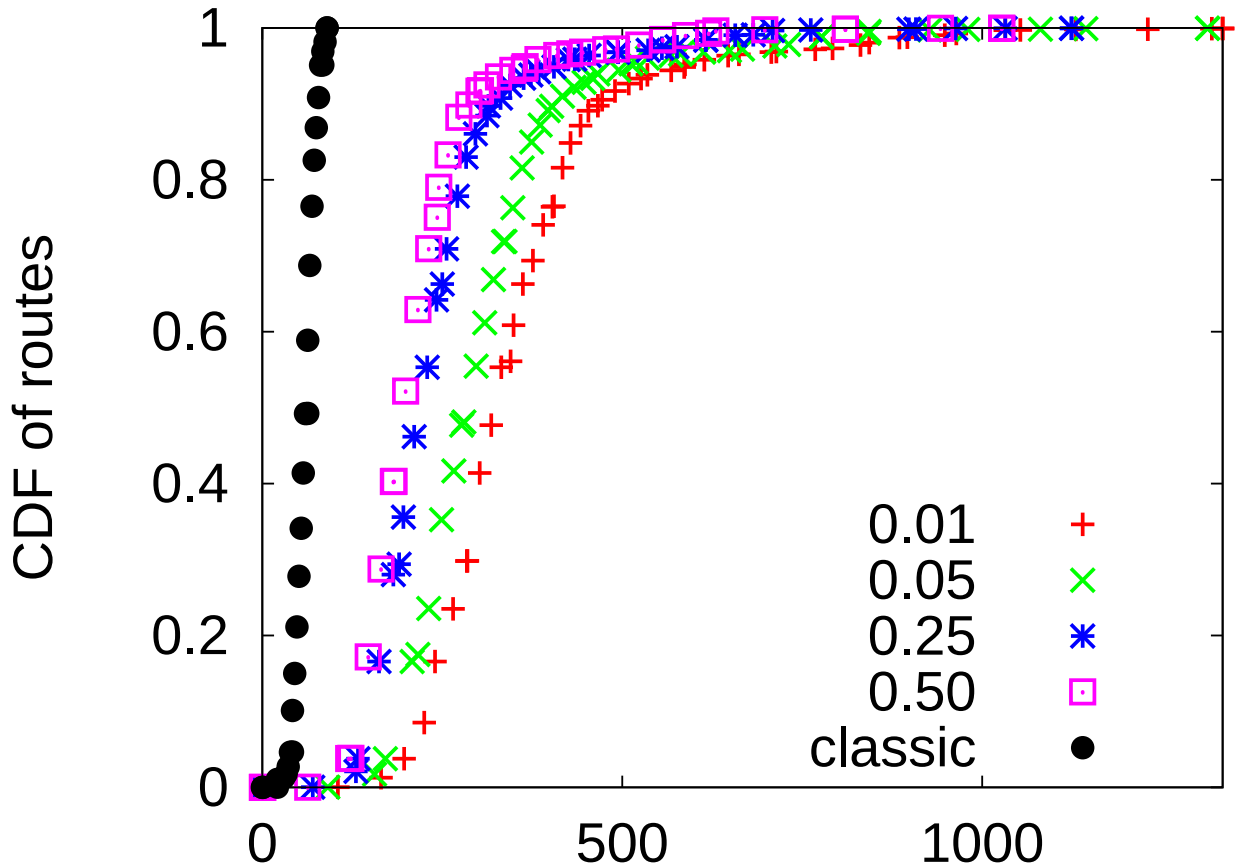
- Source: UPMC Paris Universitas
  – 44% of routes traverse a LB (prior work gives 23% – 80%)
- Destinations: 5000 randomly selected responsive to pings
- Show 1 of 4 rounds taken over 3 weeks

Link Discovery (LB paths only)



- Classic Traceroute inadequate: misses more than 10 links 25% of time!
- Failure bound not tight, $\beta^* = 0.5$ finds full multipath 94% of time

Probing Overhead (all paths)



- MDA overhead: flow key control, statistical control, retransmissions
- 20 hop route: Classic Traceroute: mean 61, worst 90
  MDA ($\beta^* = 0.5$): 216, worst 1027
  MDA ($\beta^* = 0.01$): 348, worst 1334

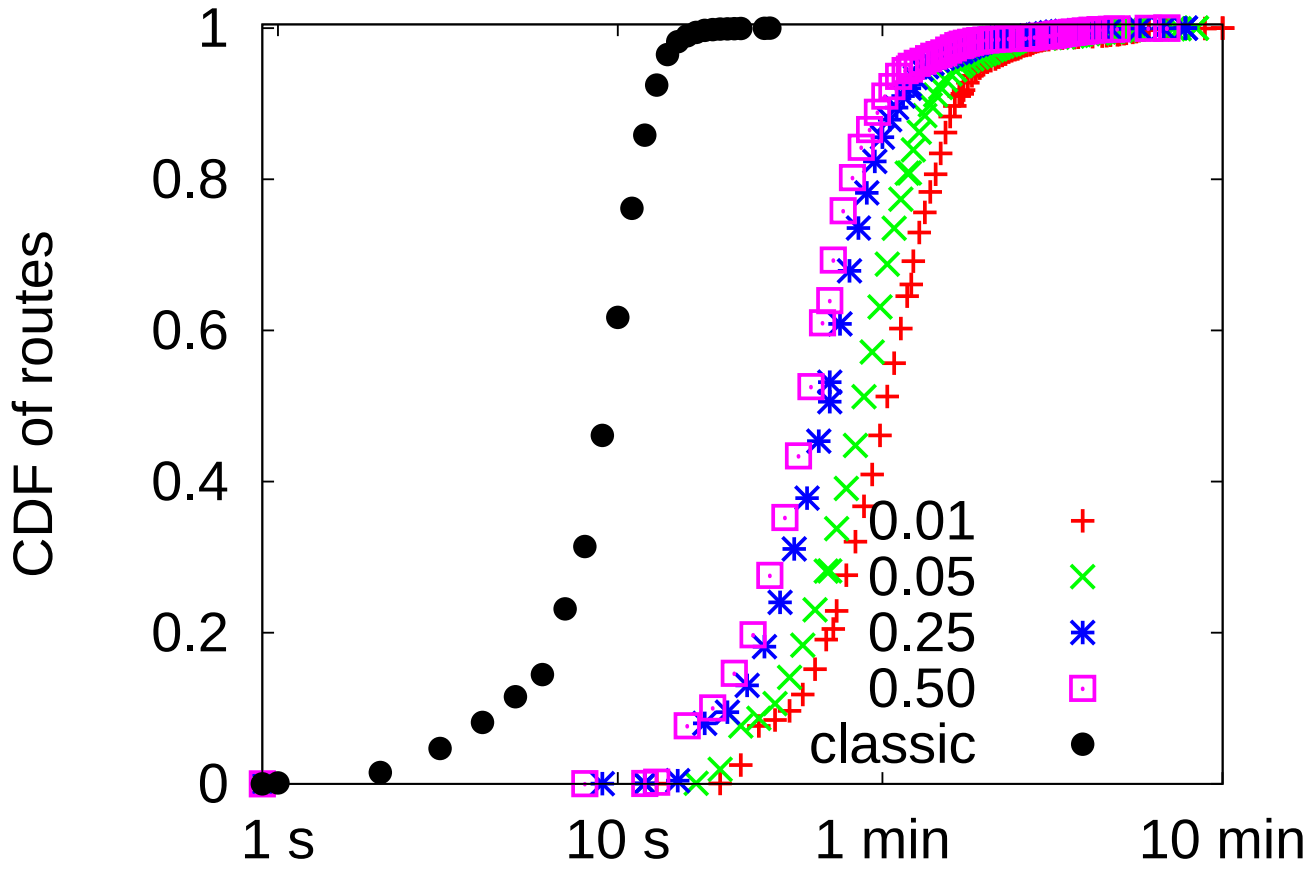
CONCLUSIONS

- Classic Traceroute totally inadequate for multipath discovery
- MDA: an adaptive algorithm to find LB generated multipaths
- For the first time: can measure multipath with confidence :
  - Failure probabilities β^* at node and graph level
    – universal bound when K unknown
    – with controlled error for target (known) topologies
  - Significance levels α_{node} at node level
- Future: reduce probe overhead with tighter bounds and and path control

CONCLUSIONS

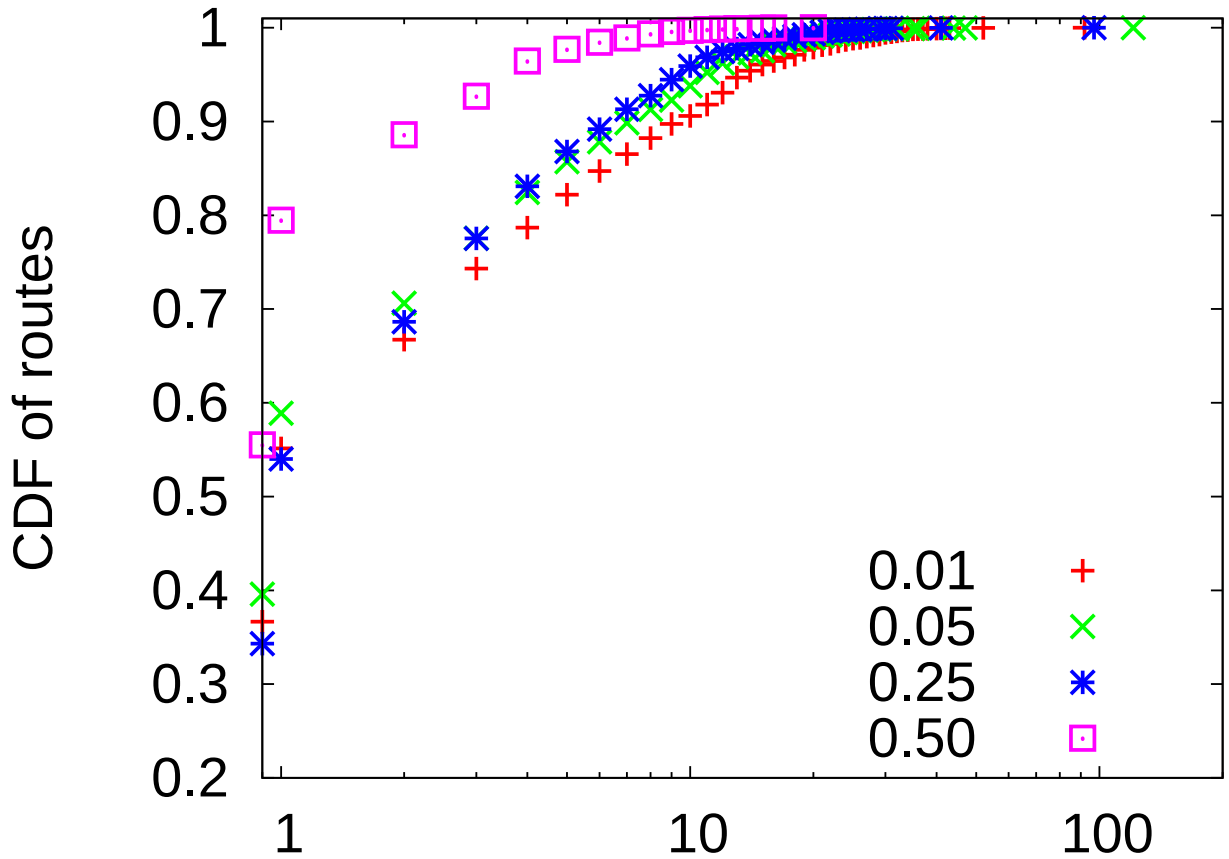
- Classic Traceroute totally inadequate for multipath discovery
- MDA: an adaptive algorithm to find LB generated multipaths
- For the first time: can measure multipath with confidence :
  - Failure probabilities β^* at node and graph level
    – universal bound when K unknown
    – with controlled error for target (known) topologies
  - Significance levels α_{node} at node level
- Future: reduce probe overhead with tighter bounds and and path control

Time Overhead (all paths)



- MDA clearly much greater in general
- Parallelism difficult but worth investigating

RETRANSMISSIONS (ALL PATHS, MDA ONLY)



- Retransmissions very common! (45% of routes for $\beta^* = 0.5$)
- Scarcely worth it, most routers reply always, or never..