ISP Driven Informed Path Selection (IDIPS)

July 2nd, 2009

http://inl.info.ucl.ac.be

Université catholique de Louvain

Agenda

- Motivation
- Informed Path Selection
  - Prediction
    - Observe
    - Predict
    - Refine
  - Ranking
  - Path Selection
- The Challenges
- Conclusion

Motivation

Traffic Engineering

- Traffic Engineering (TE) is the process of steering traffic across to the backbone to facilitate efficient use of available bandwidth between a pair of routers [1]
- In general, TE is the Art of achieving a safe and efficient transport of the flows
  - Avoid congestion
  - Minimize costs
- TE can be
  - Reactive (e.g., link *a* is congested, move to link *b*)
  - Proactive (e.g., link *a* is likely to become congested, take counter measures)

[1] Lakshman, U. and Lobo, L., MPLS Traffic Engineering, Cisco Press, 2006.

Traffic Engineering

- Today's interdomain TE is “human computing”
  - At the end of the month, move the traffic to reduce 95th percentile charge...
  - BGP `local-pref` attribute
  - AS Path prepending

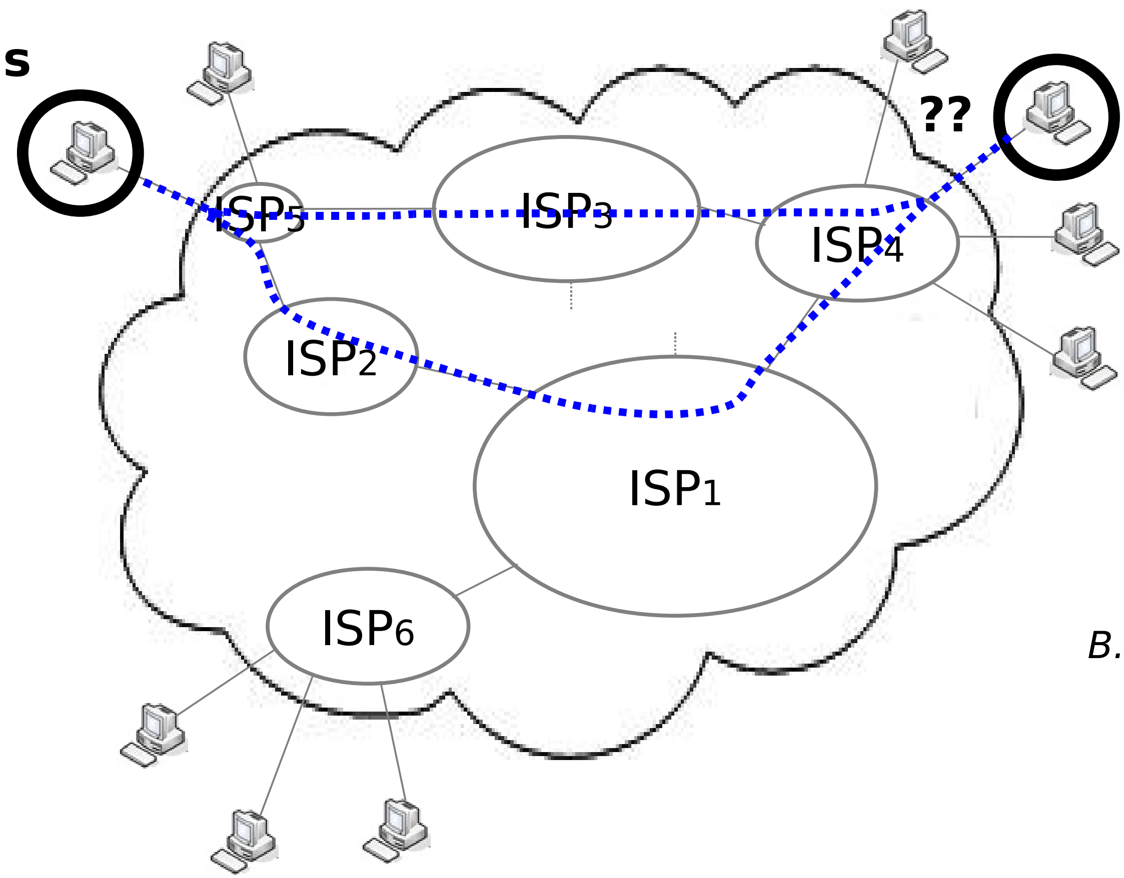
- BUT...

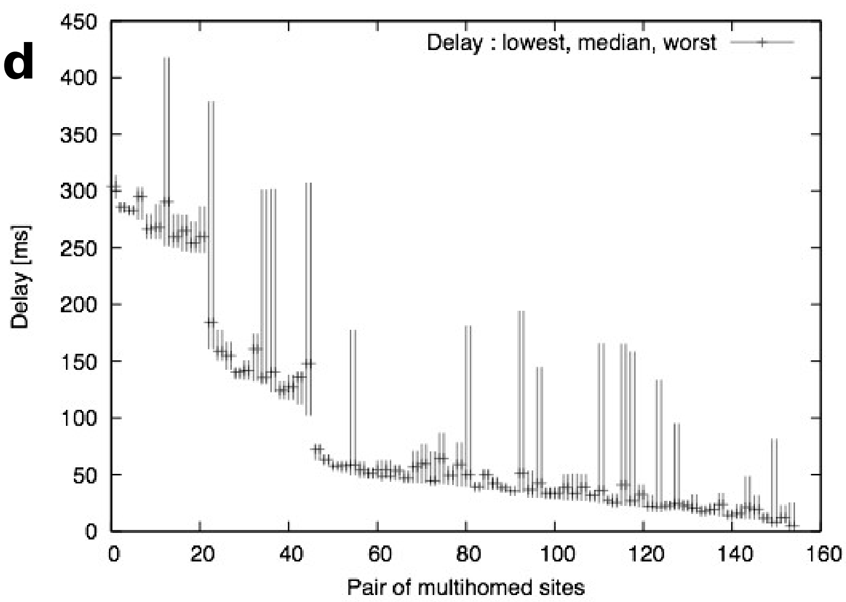
Multi-Homing (MH)

- Multi-homing implies choice among multiple feasible paths with much varying properties
  - AS-based MH: how to select the best path (ISP-based objectives)
  - Host-based MH: how to select the best path (customer-based objectives)

=> determine the best path among several:

$\{<s_1,d_1>, \ldots ,<s_1,d_n>, <s_2,d_1>, \ldots , <s_m,d_n>\}$





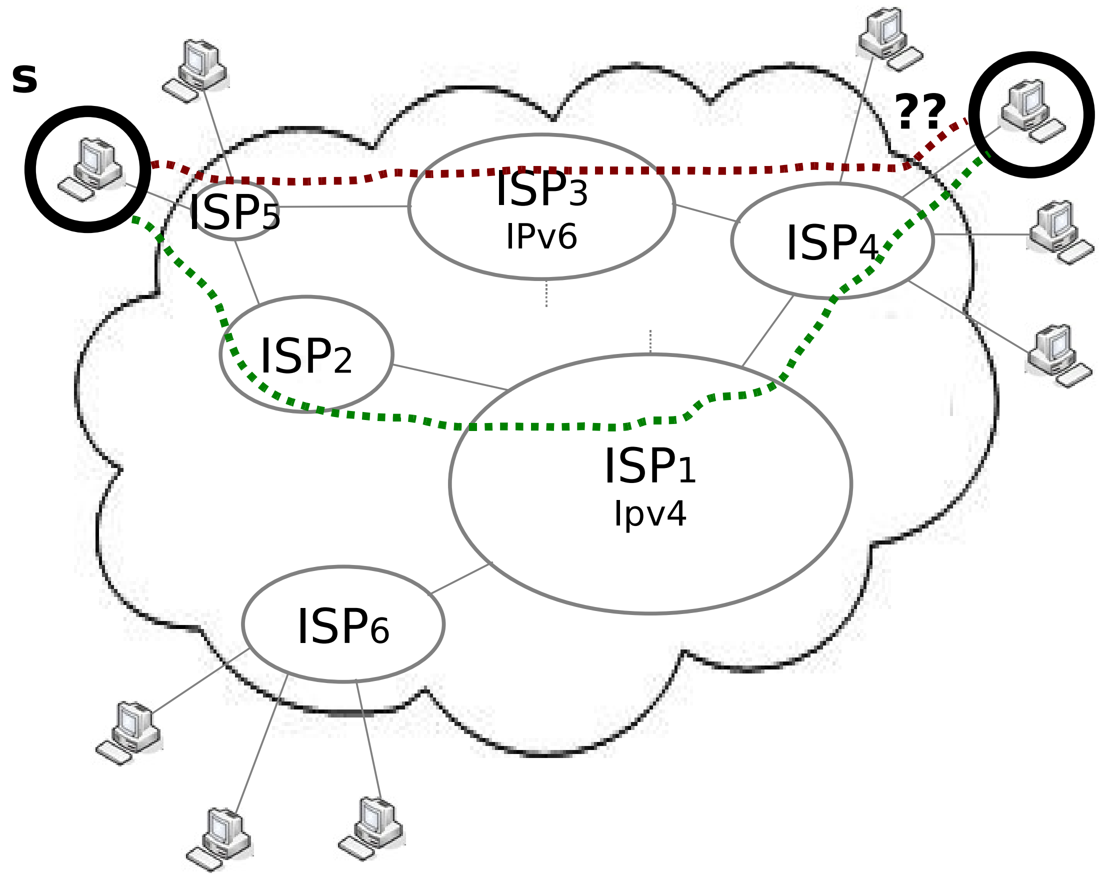
B. Quoitin et al., Evaluating the Benefits of the Locator/Identifier Separation, MobiArch'07

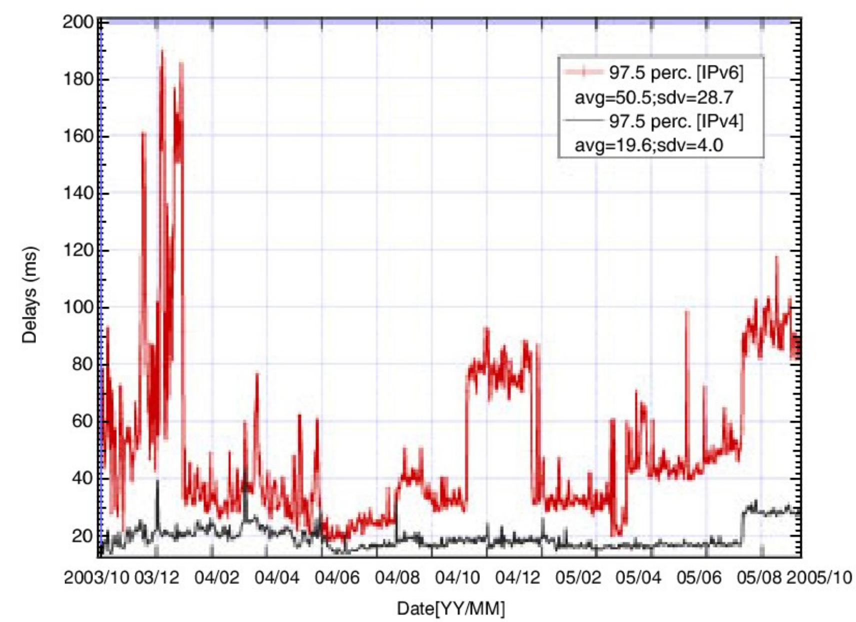
IPv4 vs IPv6 Dual Stack (DS)

- Dual stack hosts/routers will exist for many years
  - IPv4 and IPv6 performance (e.g., reliability) are not equivalent
- How to select the best stack ?
  - always prefer IPv6? RFC 3484 static selection?

=> determine the best path among several:

$\{<s_{IPv4},d_{IPv4}>, <s_{IPv6},d_{IPv6}>, <s_{IPv4},d_{IPv6}>, <s_{IPv6},d_{IPv4}>\}$





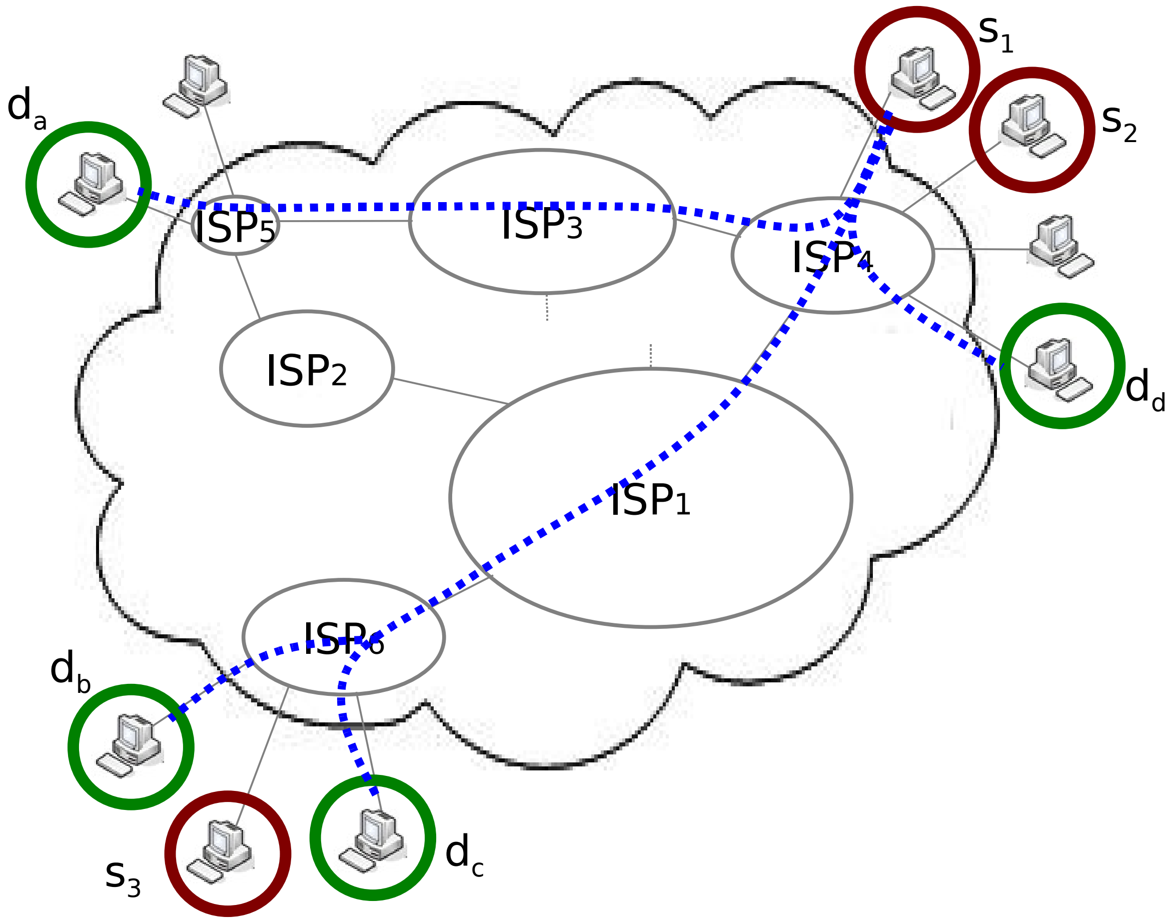
X. Zhou et al., IPv6 delay and loss performance evolution, IJCS'08

Server replicas

- How to select the best replicas
  - within set $\{d_a, d_b, d_c, d_d\}$
  - per source: s_1, s_2, s_3

=> determine the best replicaS among several:

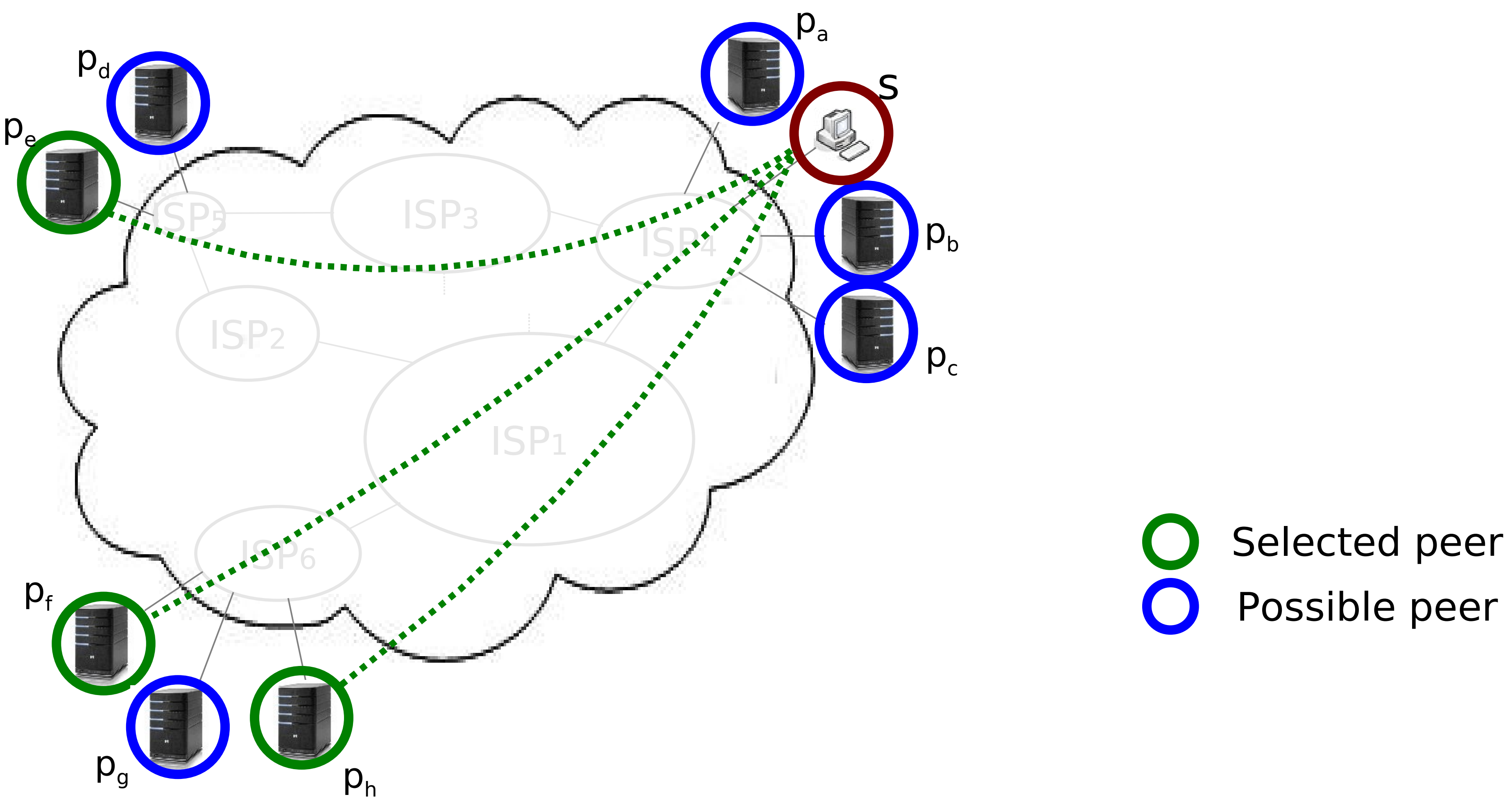
$\{<s_i,d_a>, <s_i,d_b>, <s_i,d_c>, <s_i,d_d>\}\ \forall i$



Best Peer Selection in P2P

- How to select the best peers set from the swarm
  - Example: selected peer set $\{p_e,p_f,p_h\}$ extracted from possible set $\{p_a,p_b,p_c,p_d,p_e,p_f,p_g,p_h\}$
  - per source: s_1

=> determine the best peerS among several: $\{<s,p_a>, \ldots, <s,p_h>\}$

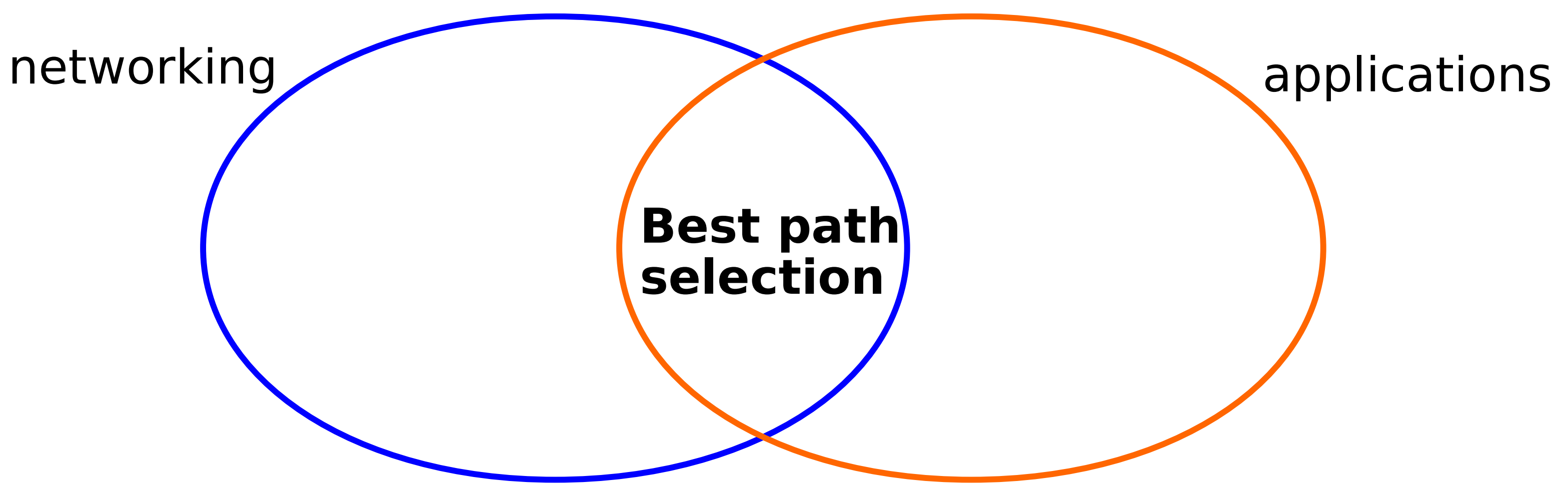


Problems are similar

- IPv4 - IPv6 DS $\in \{<s_{IPv4},d_{IPv4}>, <s_{IPv6},d_{IPv6}>, <s_{IPv4},d_{IPv6}>, <s_{IPv6},d_{IPv4}>\}$
- MH $\in \{<s_1,d_1>, \ldots ,<s_1,d_n>, <s_2,d_1>, \ldots , <s_m,d_n>\}$
- Server replication $\subseteq \{<s,d_a>, <s,d_b>, <s,d_c>, <s,d_d>\}$
- P2P Apps $\subseteq \{<s,p_a>, \ldots, <s,p_h>\}$

=> General problem $\subseteq \{<s_1,d_1>, \ldots ,<s_1,d_n>, <s_2,d_1>, \ldots , <s_m,d_n>\}$

for any s,d representation



ALL share a common problem: how to efficiently make best path selection ?

Future Internet

- TE should move from an Art to a Science
- Path performance have to be considered to sustain the Future Internet requirements

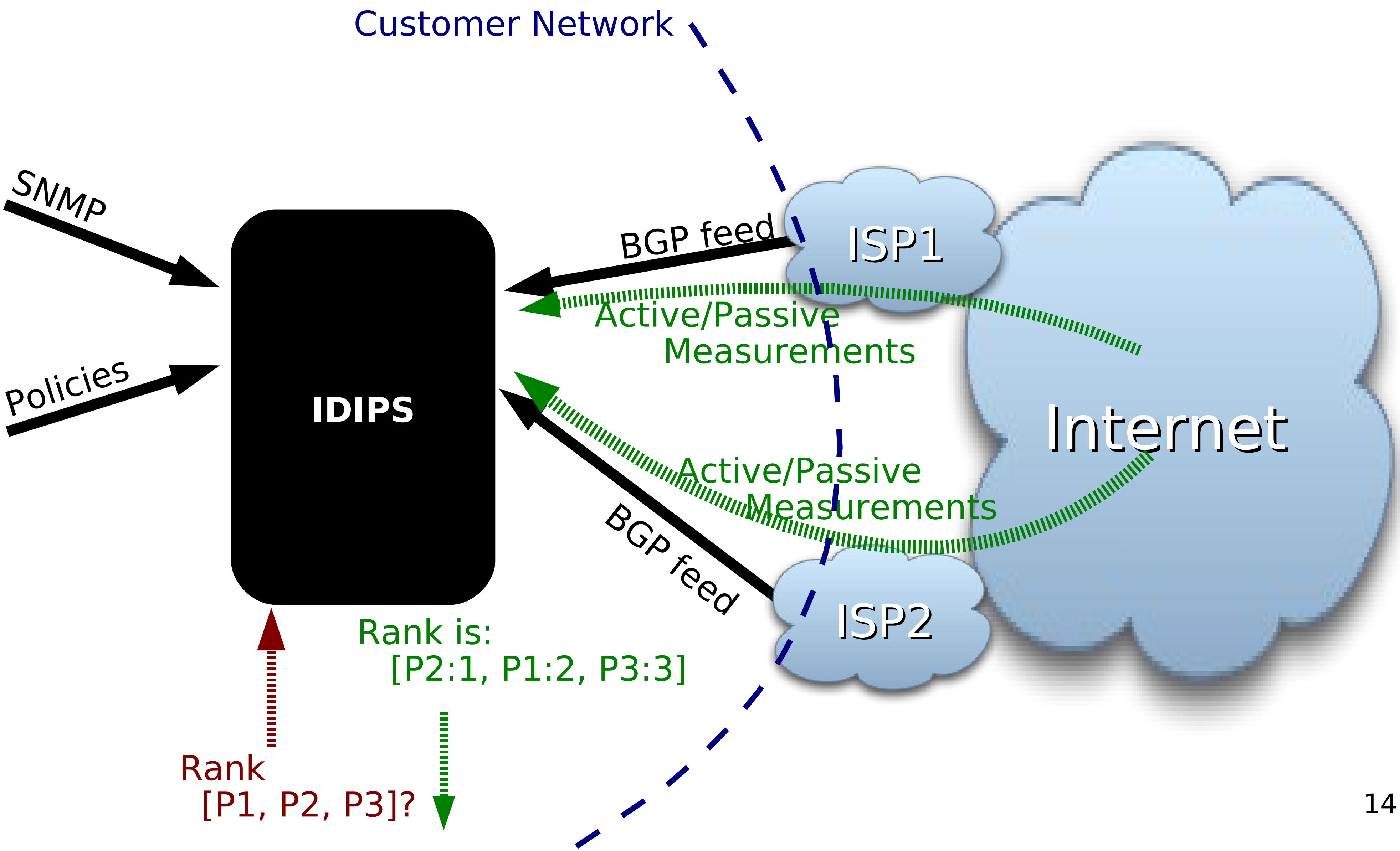
=> Informed Path Selection

Informed Path Selection

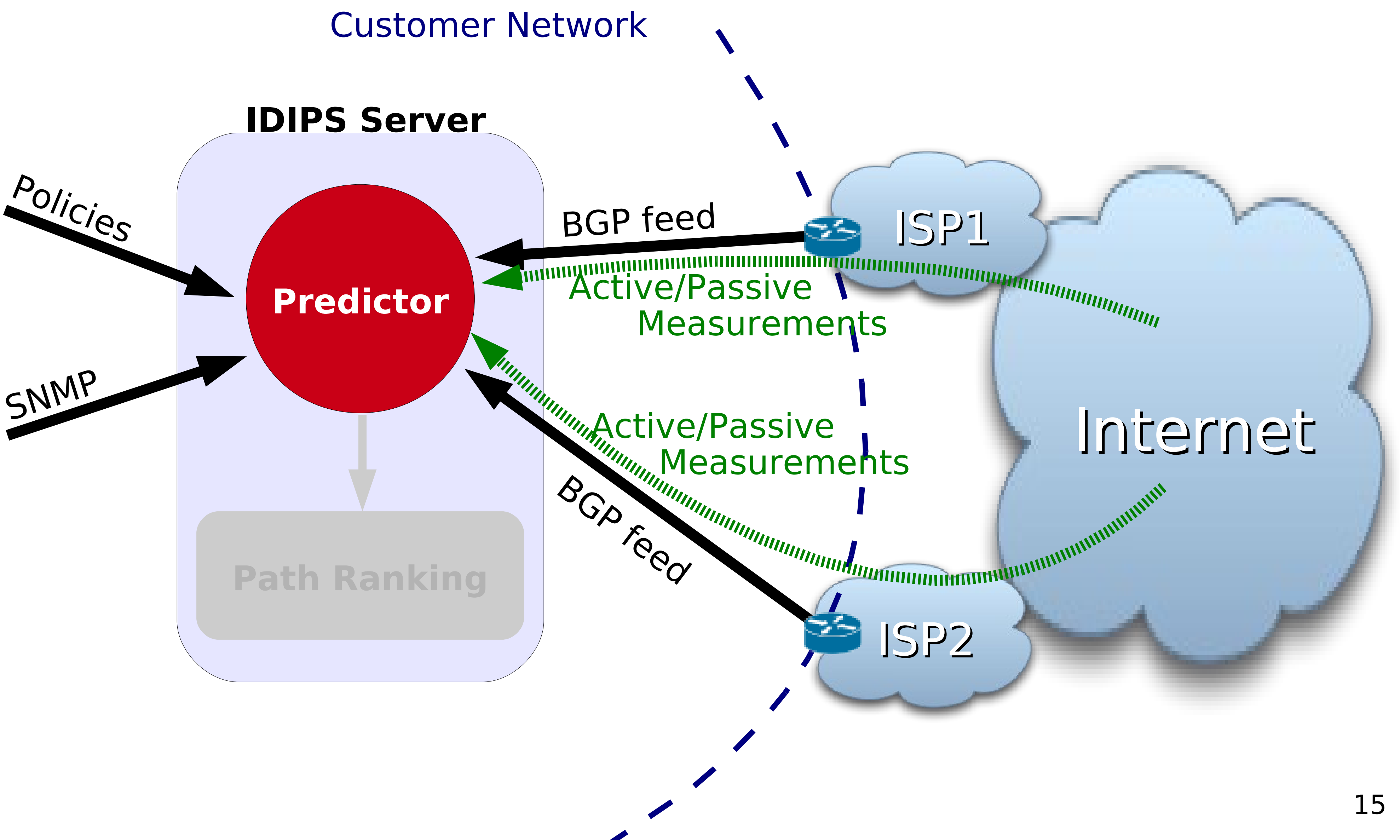
Path Selection Challenge

- We need a service able to
  - predict path performances
  - rank the paths
  - influence routing decisions
- This system would be
  - auto adaptive
  - flexible
  - iteratively deployable

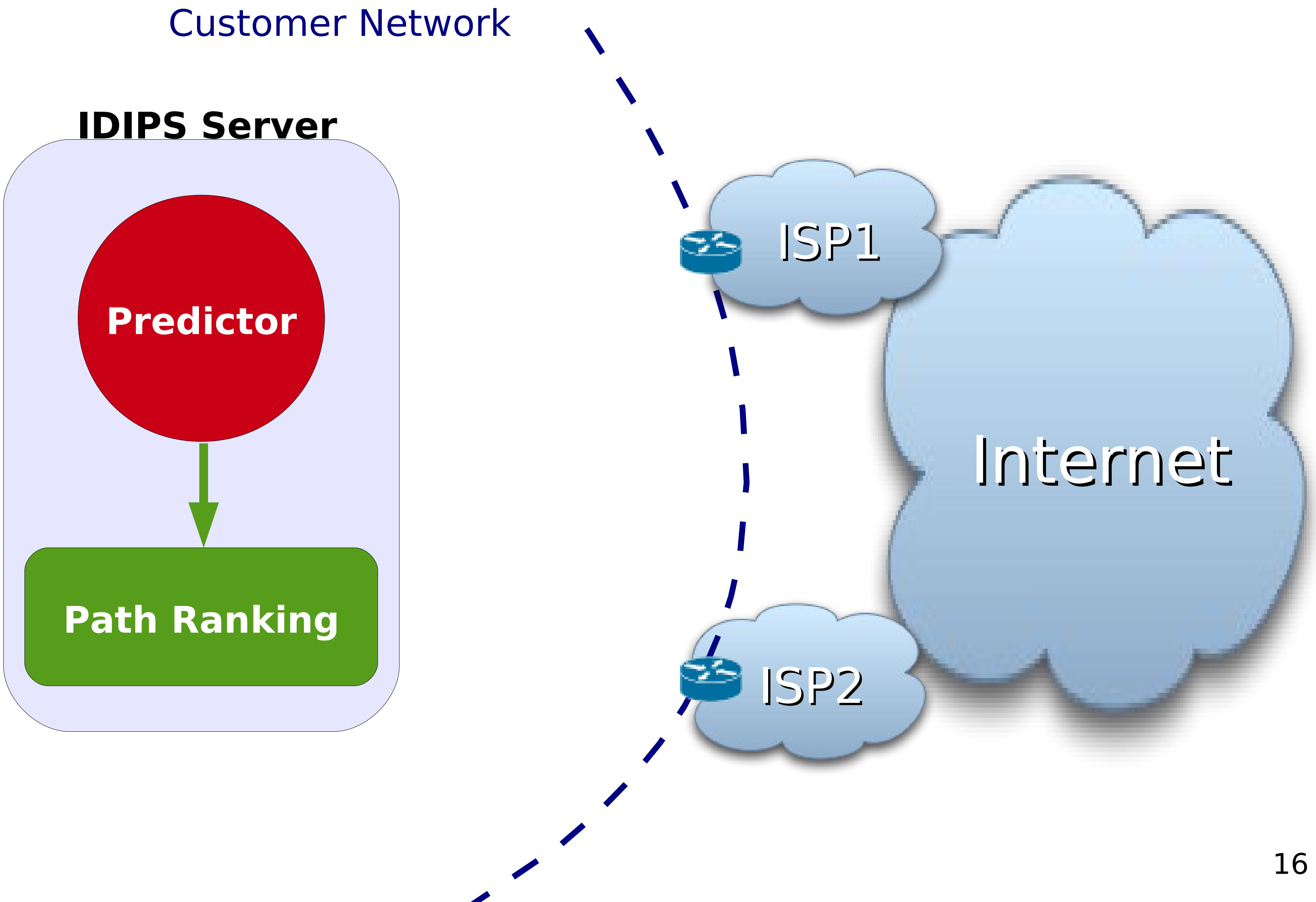
IDIPS: ISP-Driven Informed Path Selection



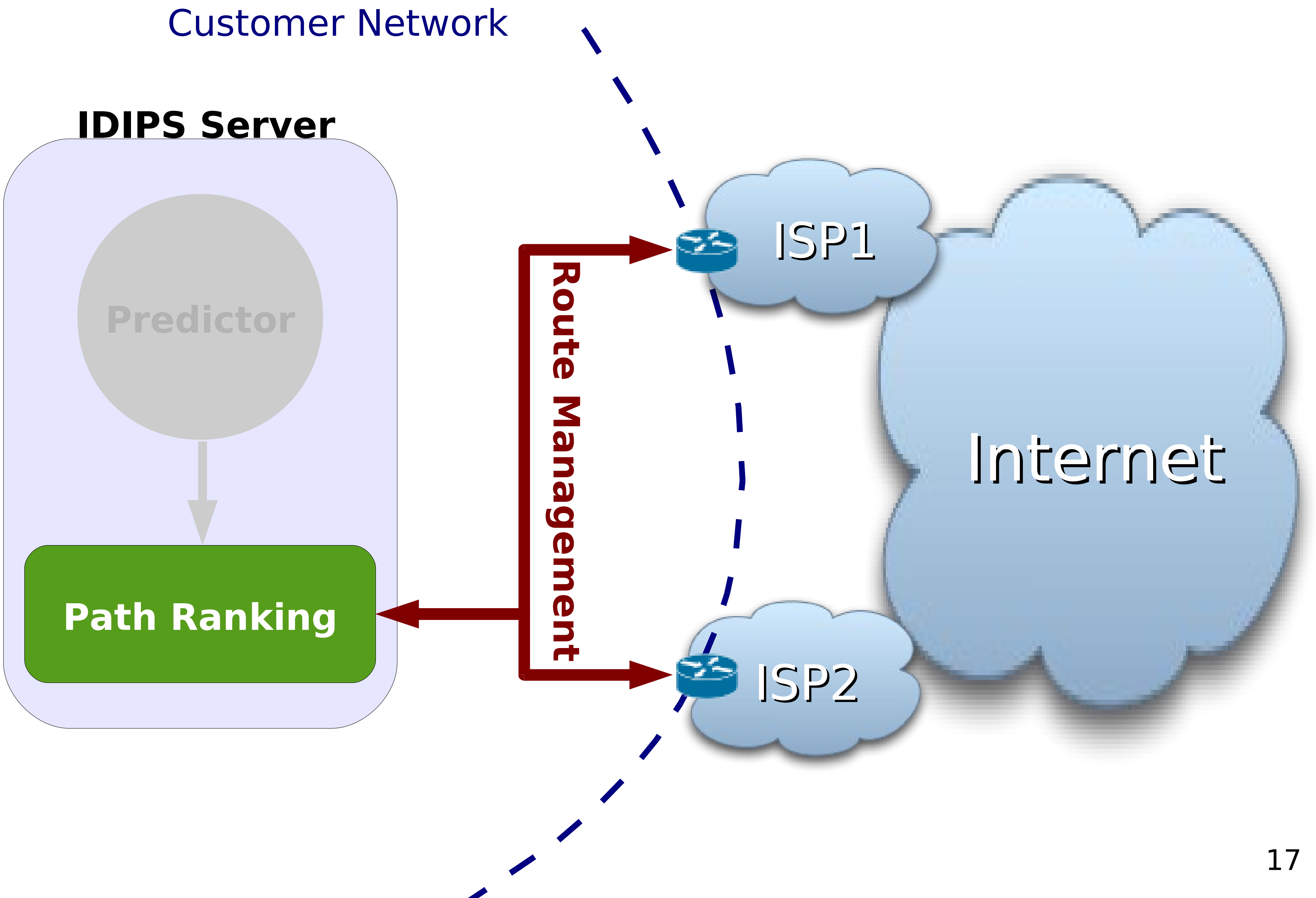
IDIPS Components
(Predictor)



IDIPS Components
(Path Ranking)

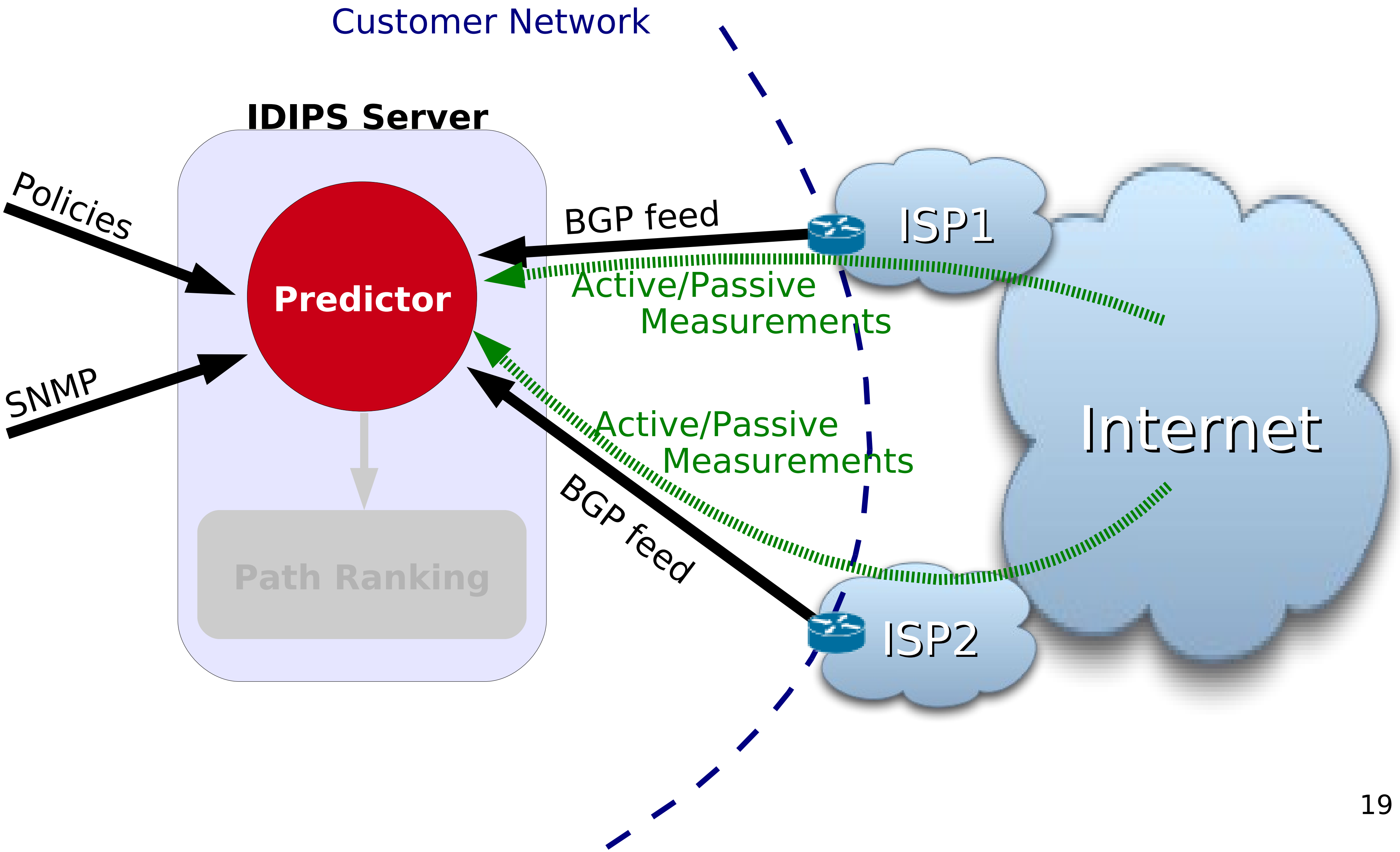


IDIPS Components
(Route Management)



Inside IDIPS
(Path Performance Prediction)

Inside IDIPS



Path Performance Prediction

1. Observe the performance of the paths

2. Predict the future performance of the paths

3. Refine the predictions

Observe

Active vs Passive Measurements

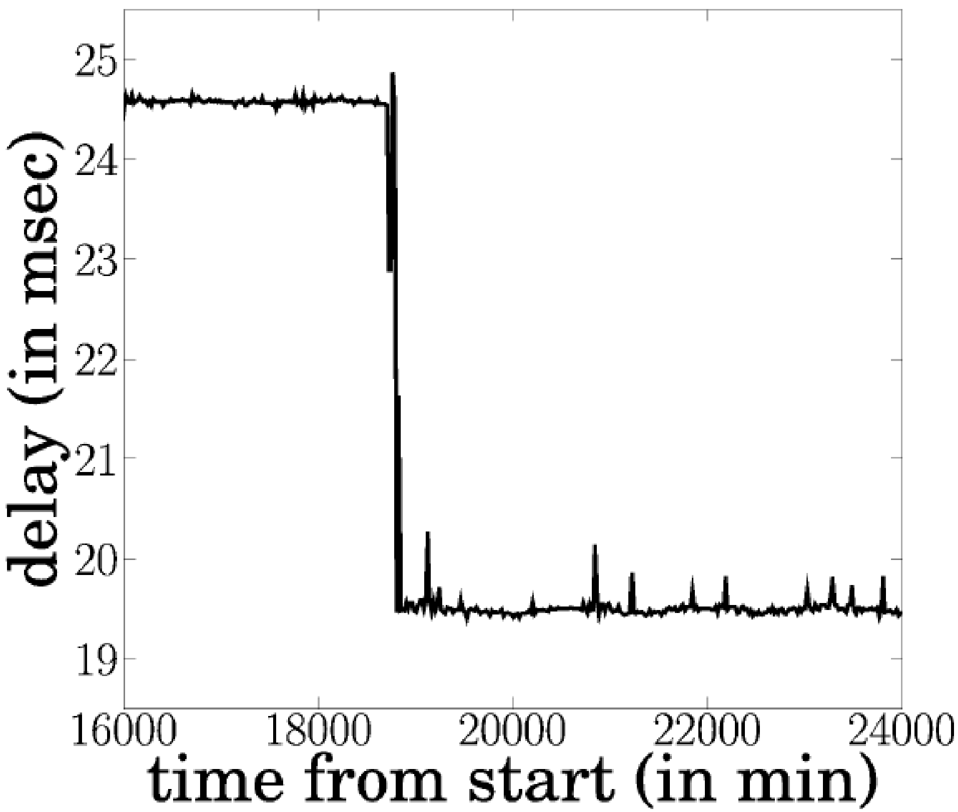
- Passively measuring the traffic is not sufficient
  - Measures only the paths carrying traffic
    - NetFlow
- Actively measuring all the paths is not feasible
  - Does not scale

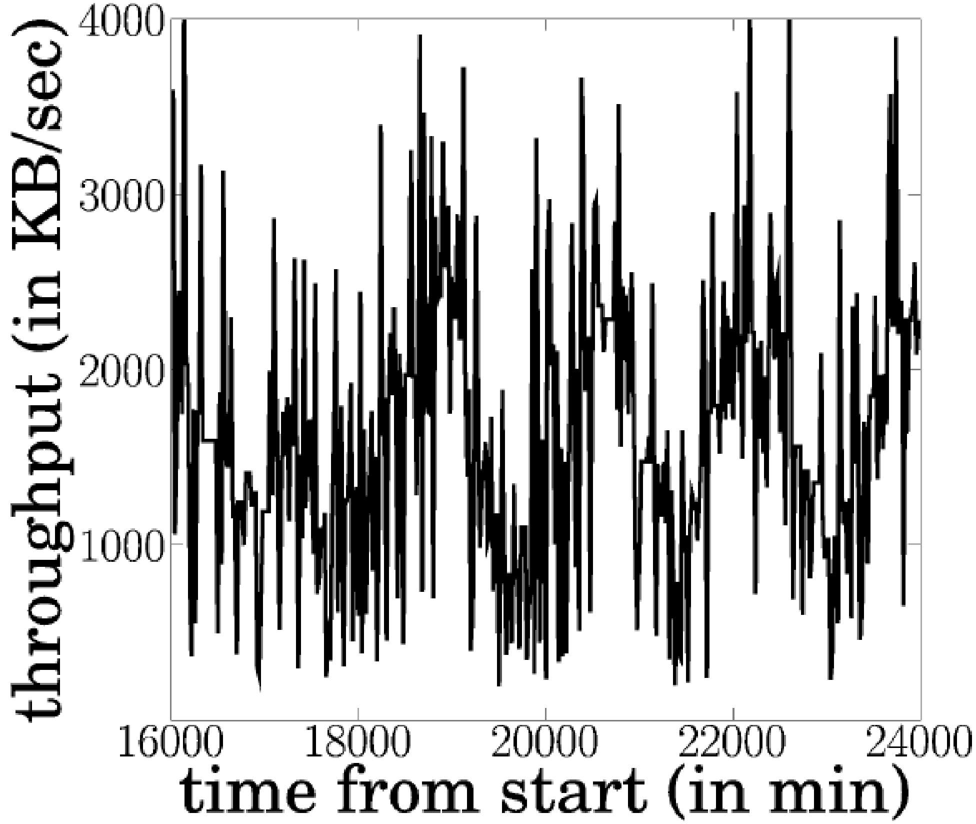
Active vs Passive Measurements

- Passively measure all the traffic and detect abnormal behavior (cf. Kavé's talk)

- Actively measure the most important destinations (and the paths to them)
  - Manually configured (e.g., VoIP)
  - Dynamically (e.g., cover 90% of the traffic)
  - Confirmation of an anomaly

Typical observation

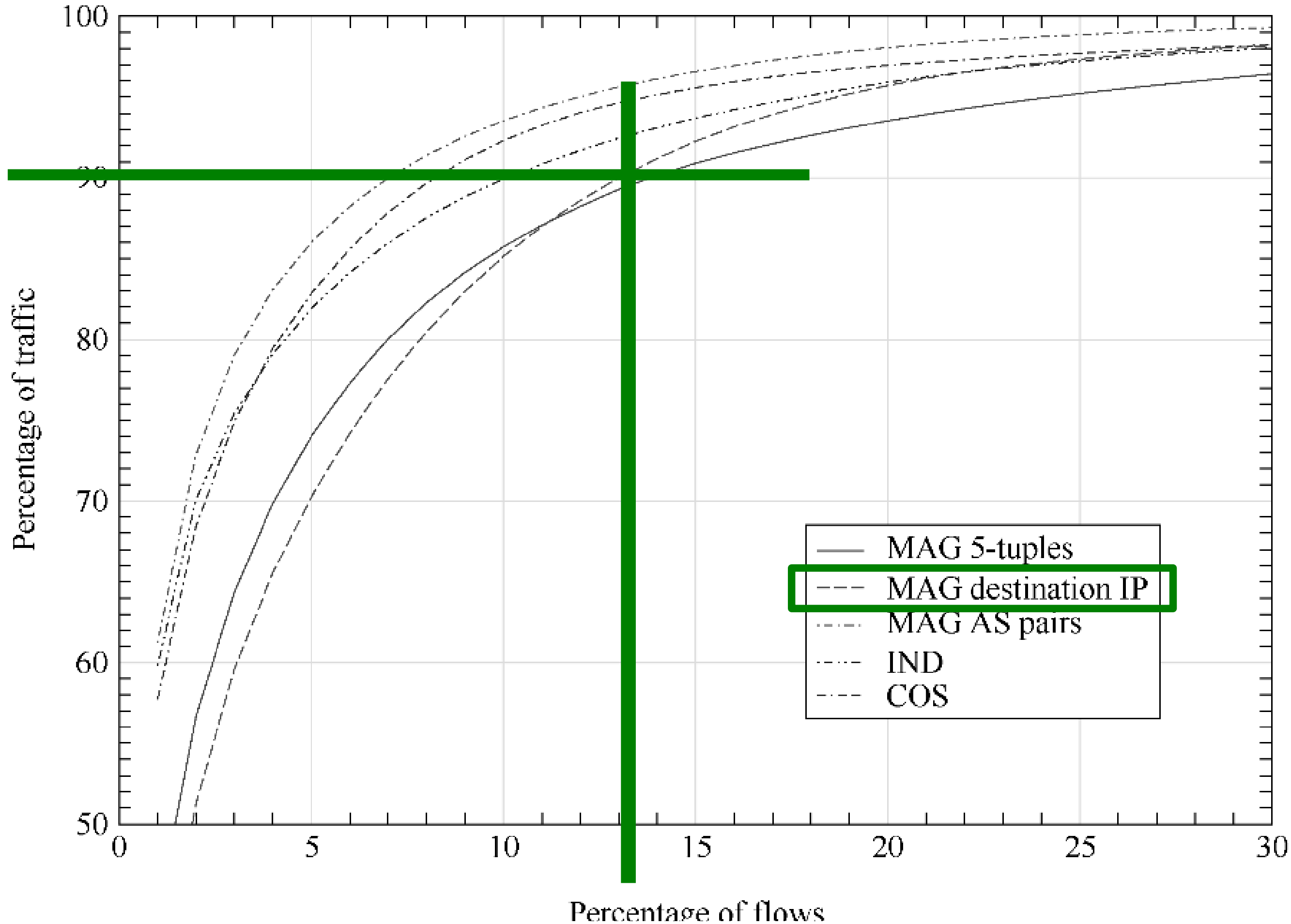




Pablo, J. et al., A Comparative Study of Path Performance Metrics Predictors. ACM SIGMETRICS Advanced Learning for Networking Workshop'09

Reduce the number of measured destinations

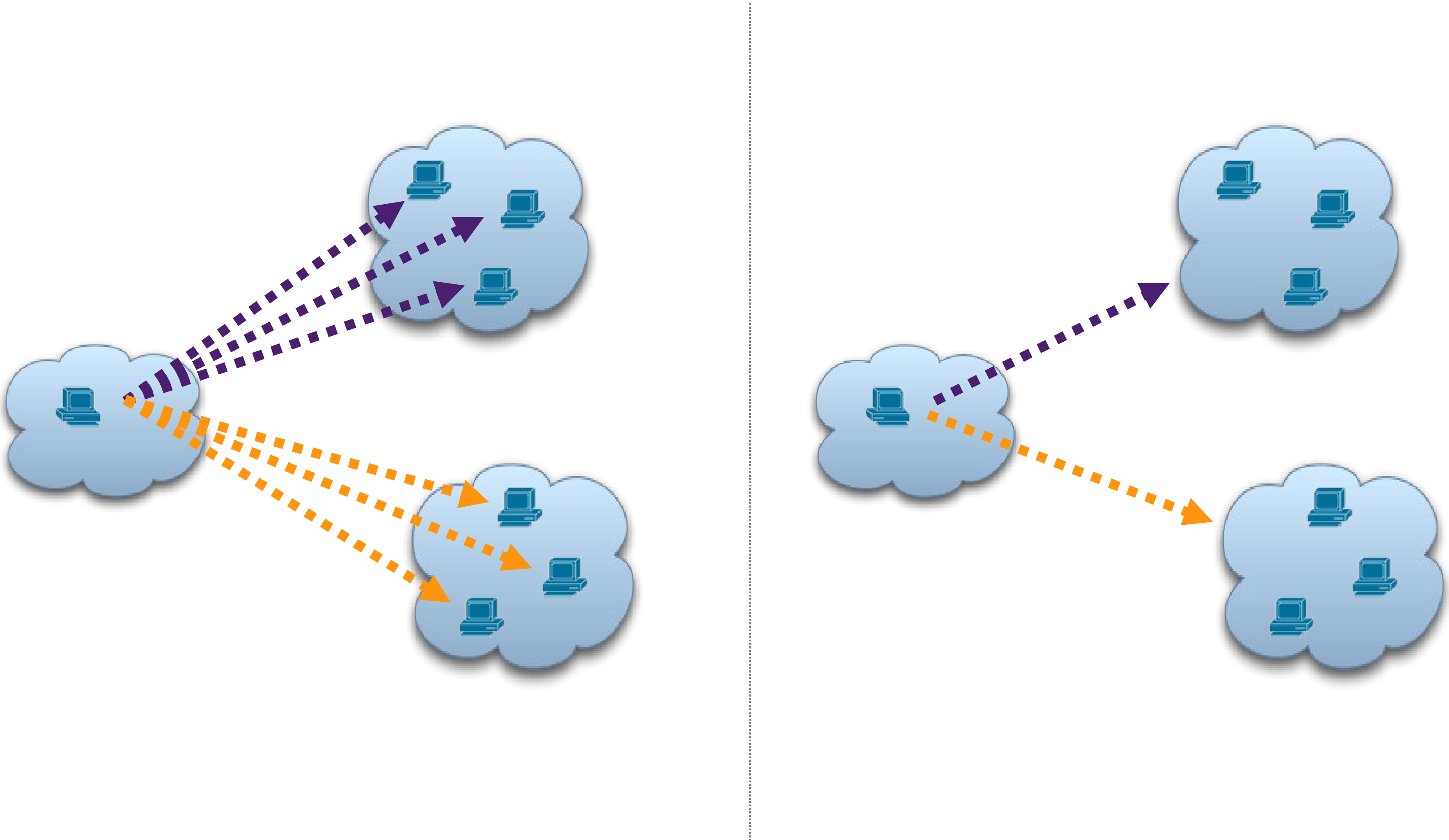
- Consider the top talkers



C. Estan and G. Varghese, *New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice*, ACM TCSystems 21(3) 2003.

Reduce the number of measured destinations

- Group the destinations into clusters

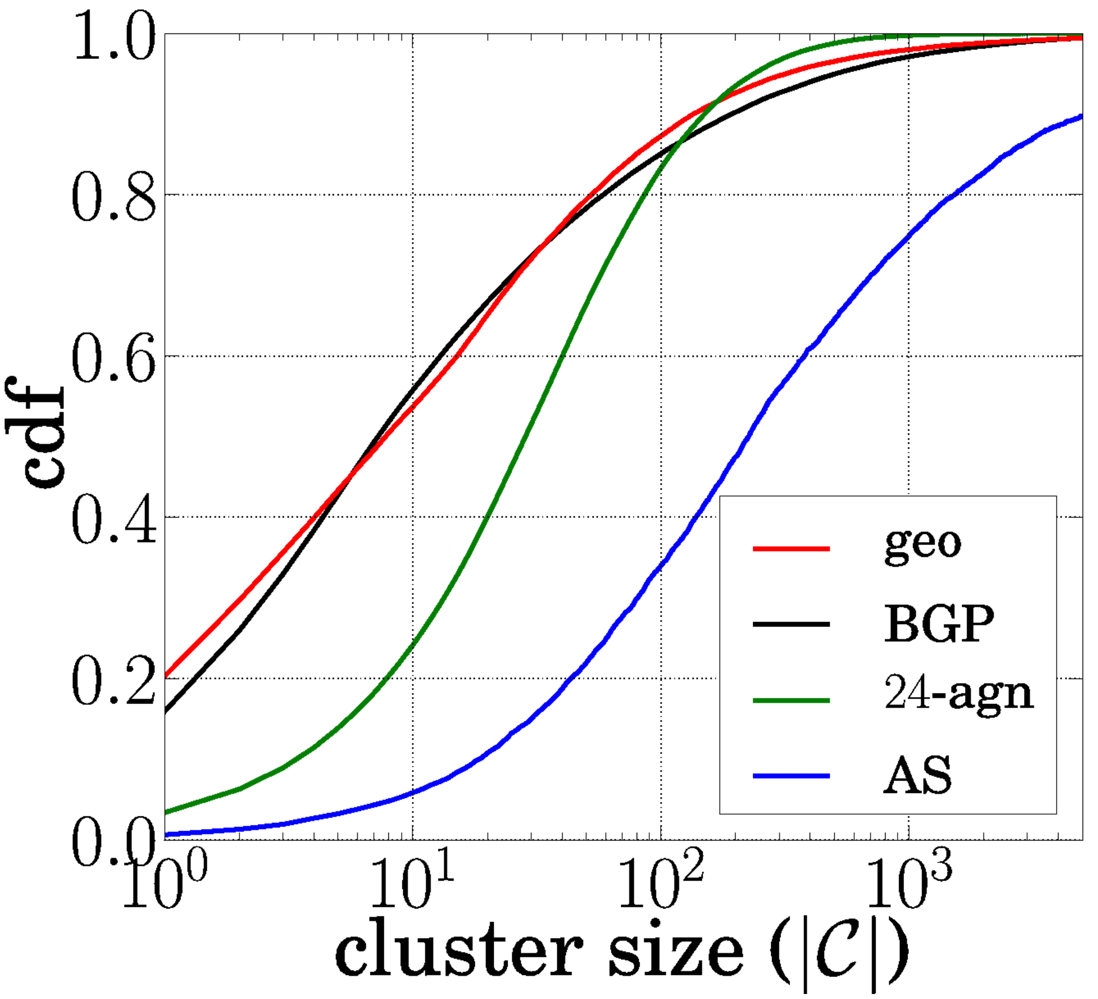


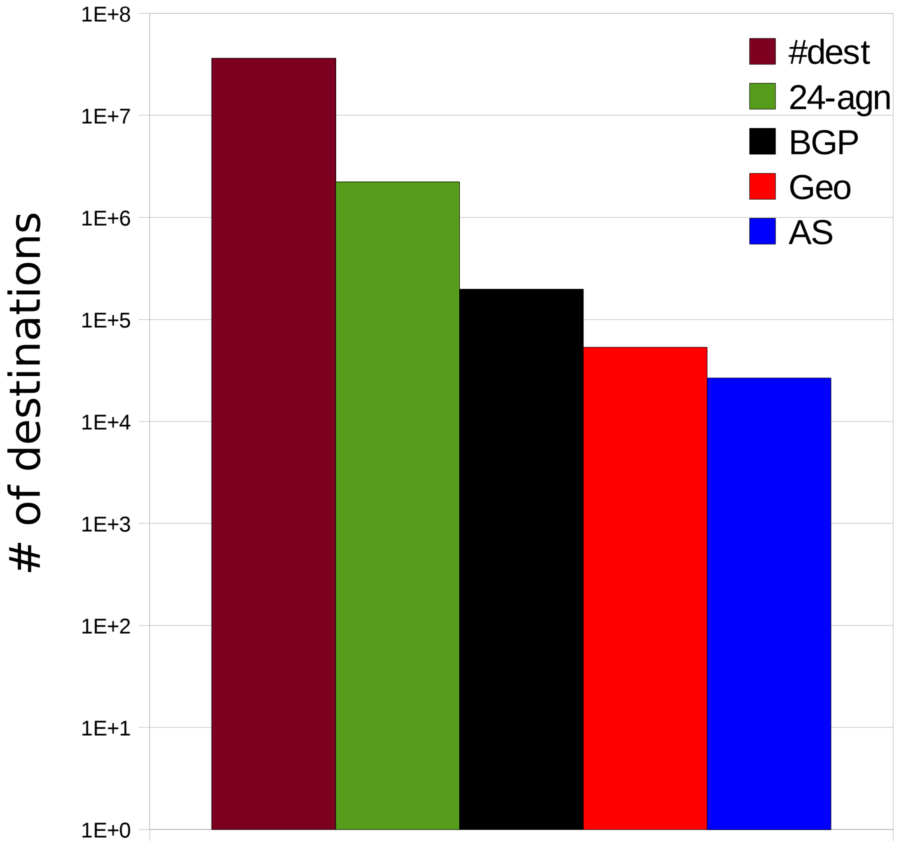
Clustering Techniques

- Geographic Clustering
  - Group destinations by city
- n-agnostic Clustering [1]
  - Group nodes by */n* prefixes
- AS Clustering [2]
  - Group nodes by autonomous systems
- BGP Clustering [3]
  - Group nodes by longest-match BGP prefix

[1] Szymaniak, M. et al., Practical large-scale latency estimation. Computer Networks'08
[2] Krishnamurthy, B., Wang, J., Topology modeling via cluster graphs. IMW'01
[3]Krishnamurthy, B., Wang, J., On network-aware clustering of web clients. ACM SIGCOMM'00

Effective Reduction with Clustering*

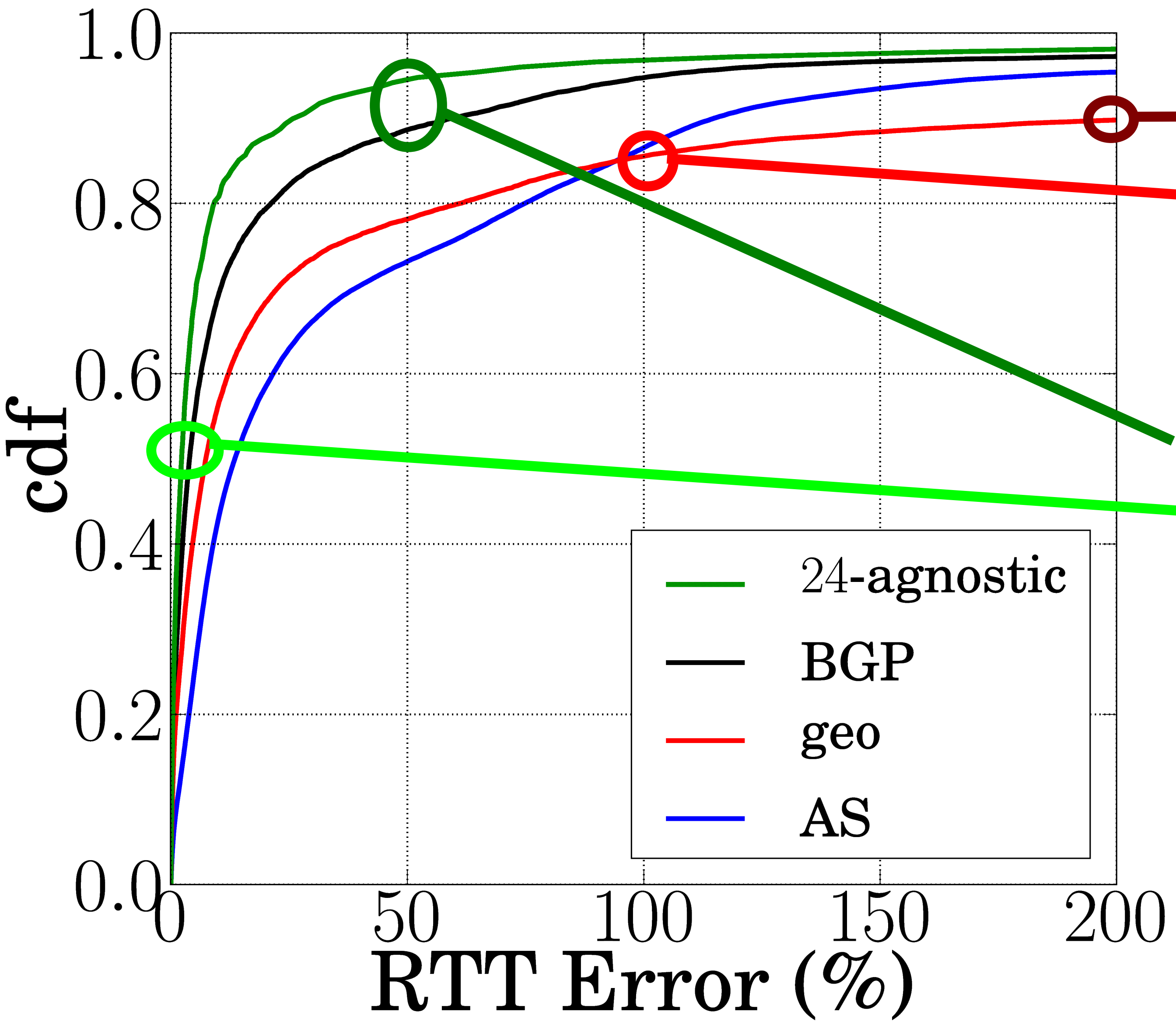




At least 45% of the clusters cover more than 10 nodes

* 1 month campus traffic, 7.45TB of outgoing traffic

Impact of Clustering technique on accuracy*



10% with more than 200% error

15% with more than 100% error

90% with less than 50% error

50% with less than 10% error

Geographic, AS 😐

n-agnostic, BGP ☺

$$e_{ij} = \frac{|m_{ij} - \widehat{m}_{ij}|}{m_{ij}}$$

* 1 month Archipelago trace

Predict

Machine Learning Problem

- Performance prediction can be seen as a Machine Learning problem
  - Input:
    - Observed performance
  - Output
    - Prediction of the future performance
  - Challenge
    - Find a model that fits with the reality
    - Tune model' parameters

Data preprocessing

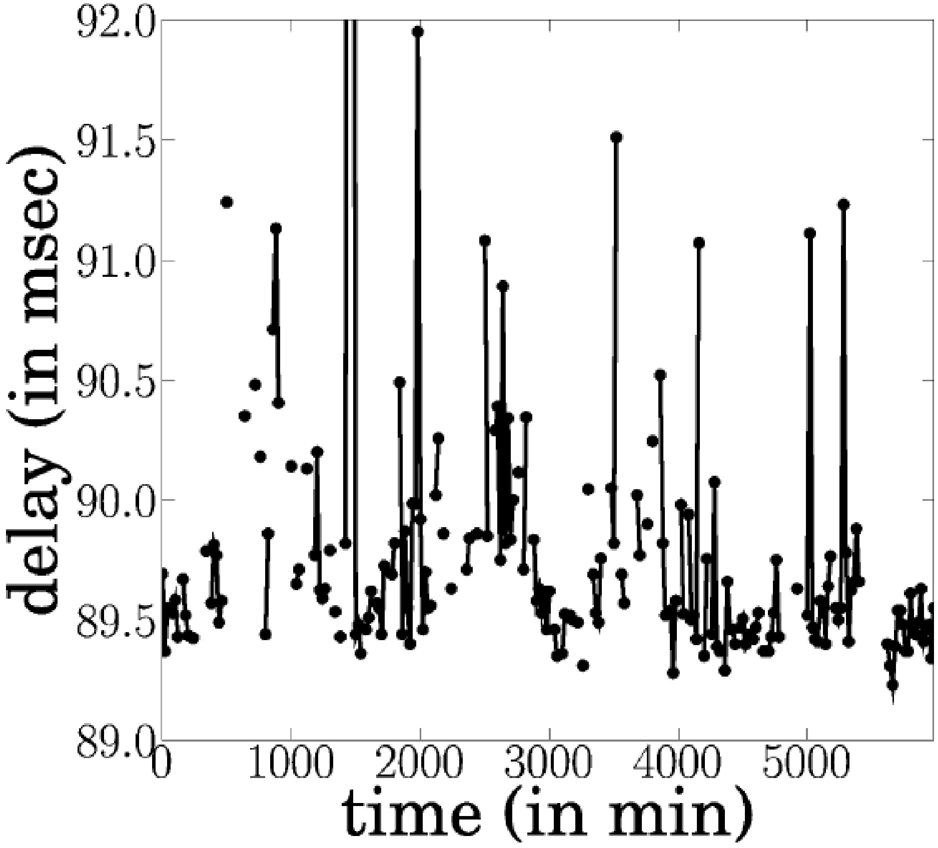
- Sometimes, observed data contain “gaps”
  - Transient failures, packet loss
- Data imputation (smooth fit):
  - Average,
  - Median,
  - *k*-nearest neighbor

$$\hat{y}_t = \frac{1}{k} \sum_{j=-k/2}^{k/2} y_{j-k}$$

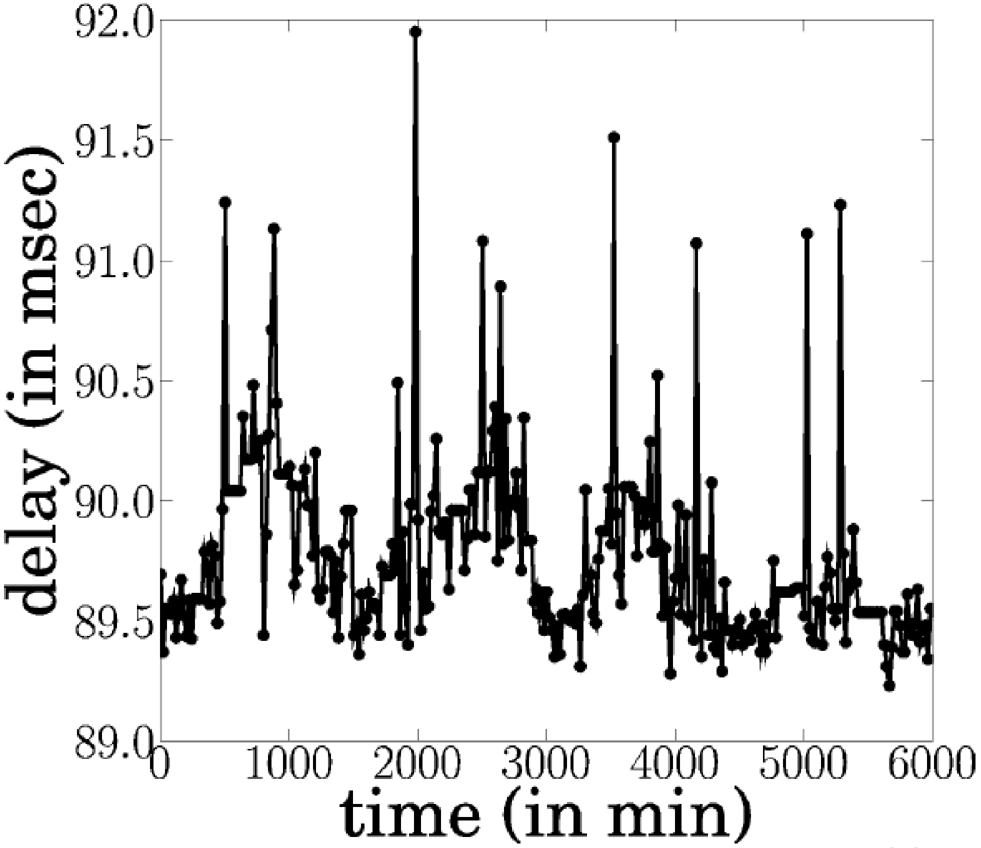
Data preprocessing

- Example of delay gapped data with *6-nearest neighbor* imputation

Before



After



Pablo, J. et al., A Comparative Study of Path Performance Metrics Predictors. ACM SIGMETRICS Advanced Learning for Networking Workshop'09

Time Series Analysis

- Time Series Analysis: predict the metrics given a series of past observations
  - In the past, a time series of a particular metric has been seen, the future values of this metric could be predicted
  - Given a set $D = \{y_0, \ldots, y_t\}$ of previous measurements
  - Try to calculate y_{t+k} for any given k, given D

Autoregressive Moving Average (ARMA)

- Autoregressive Moving Average (ARMA): try to predict future values of the time series, by making a linear combination of previous values

$$y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \phi_j \epsilon_{t-j} + e_t$$

- Other techniques like *Kalman Filters* or *Support Vector Regression* are being studied

Autoregressive Moving Average (ARMA)

- ARMA (p,q)

Moving Average (MA)

$$y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \phi_j \epsilon_{t-j} + e_t$$

Auto Regression (AR)

White Noise

$e_t \sim N(0, \sigma^2)$

Autoregressive (AR)

- AR(p)

$$\sum_{i=1}^{p} \alpha_i y_{t-i}$$

- *p* give the number of past observations to remember

- ARMA (1,0)
  - $y_t = \alpha\ y_{t-1} + e_t$

Moving Average (MA)

- MA(q)

$$\sum_{j=1}^{q} \phi_j \epsilon_{t-j}$$

with $\epsilon_i \sim N(0, \sigma^2)$

Autoregressive Moving Average (ARMA)

- ARMA (p,q)

$$y_t = \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{j=1}^{q} \phi_j \epsilon_{t-j} + e_t$$

- What order for AR? For MA? (e.g., AIC)
- α_i parameter? Φ_j parameter? (e.g., MLE)

Refine

Performance Index

- Percent Mean Absolute Deviation (PMAD):

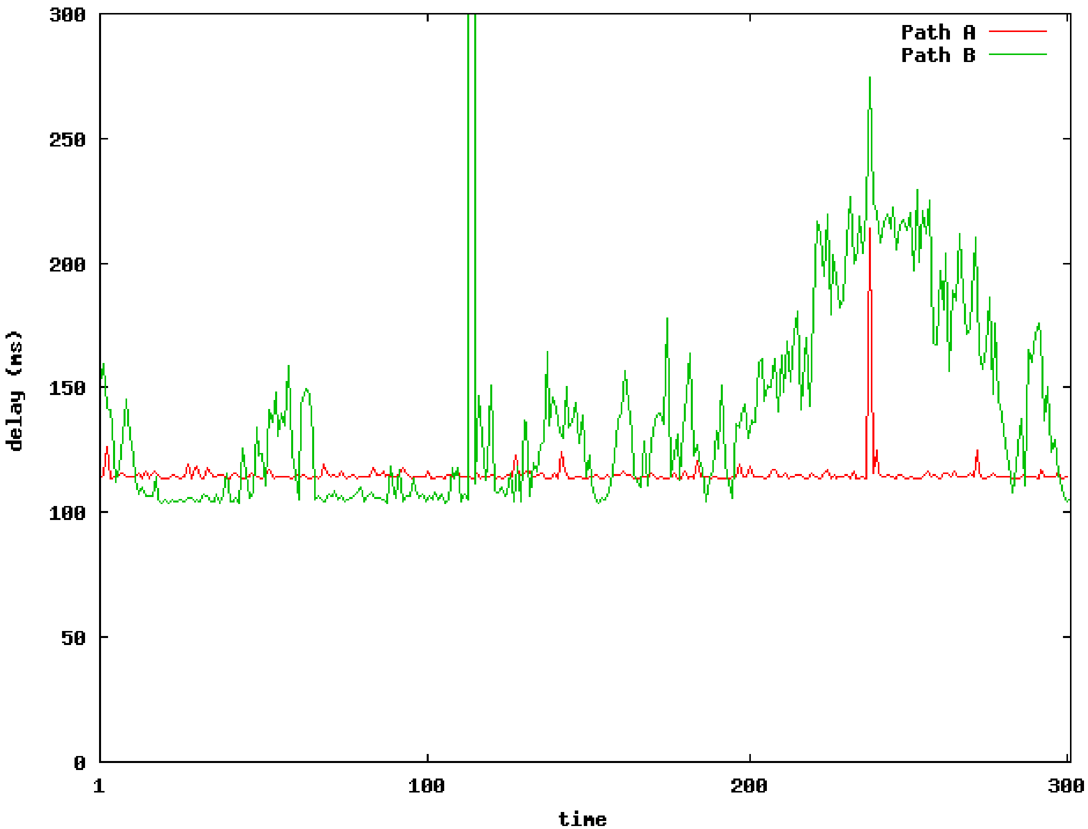
$$PMAD = \frac{\sum_{i=1}^{N} |e_i|}{\sum_{i=1}^{N} |y_i|}$$

- Where $e_i = \hat{y}_i - y_i$ is the difference between the predicted and the actual value

- Used to tune ML learning model' parameters

Sampling frequency

- Some observed path are stable, others are less
  - How to adapt the sampling frequency?



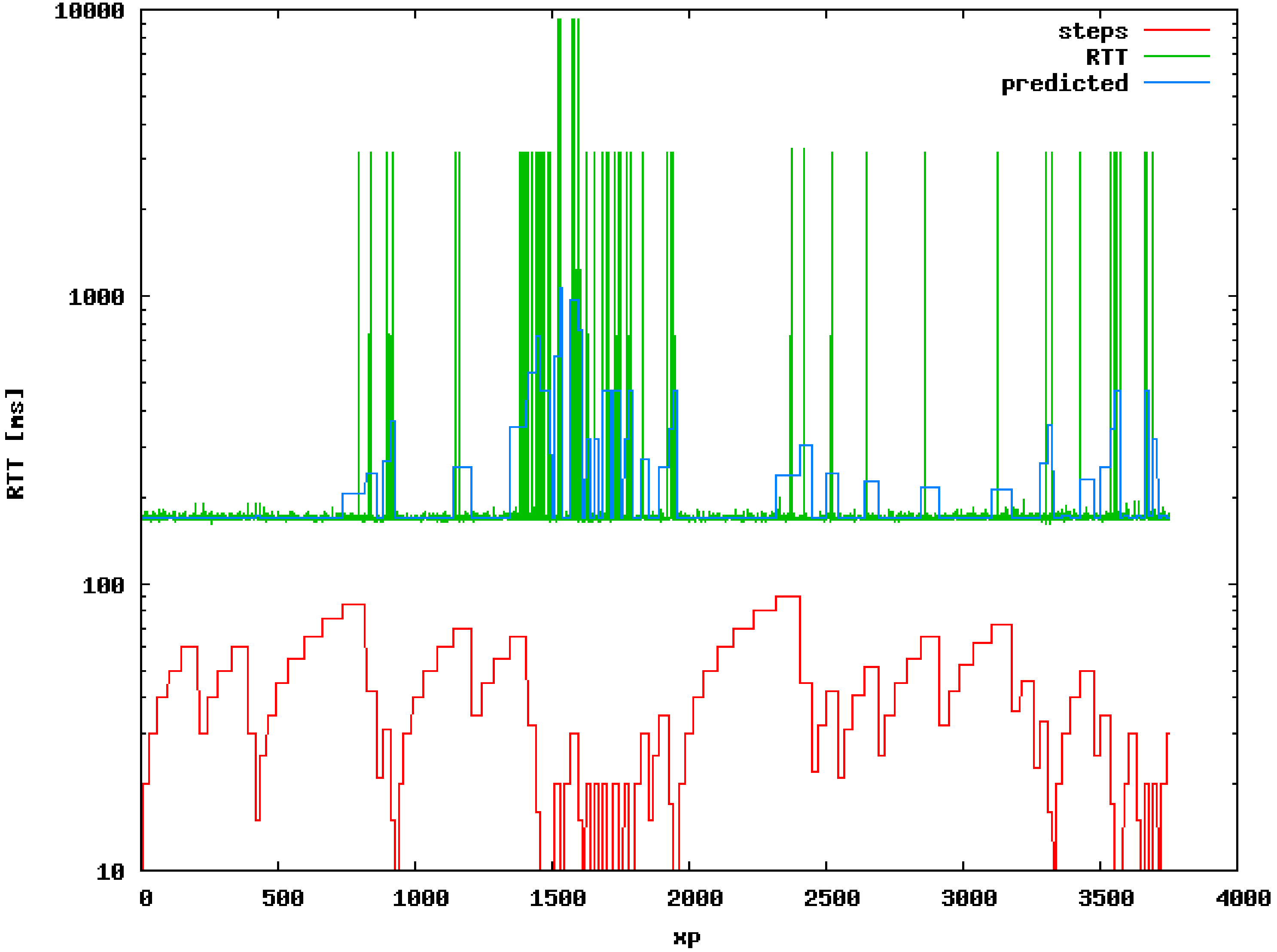
Sampling frequency

- Let P, the sampling period

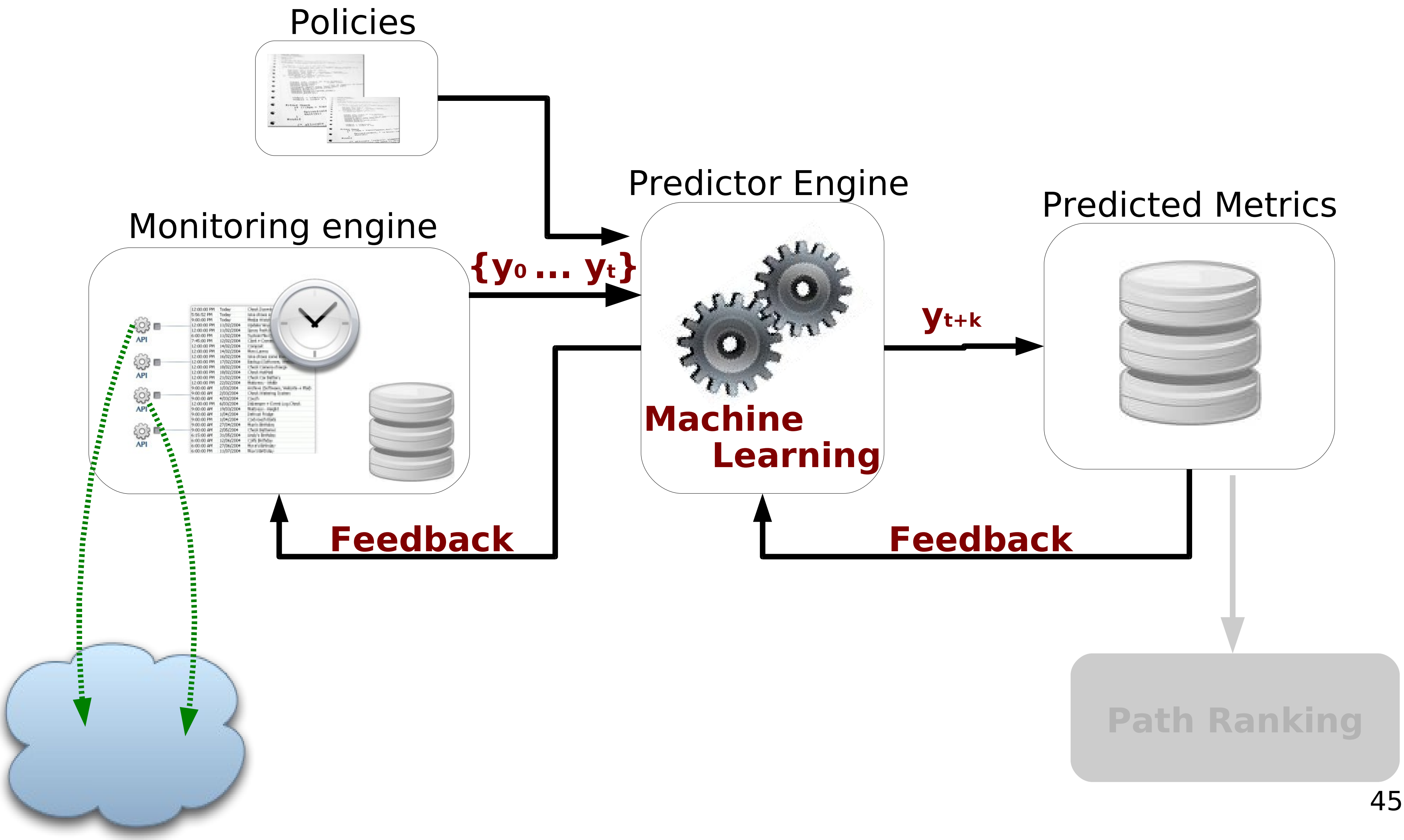
```
if prediction error > threshold then
    P := P / 2
Otherwise
    P := P + 1 bin
```

- P >= minimum threshold
  - Limit the maximum frequency to limit the overhead
- P <= maximum threshold
  - Limit minimum frequency to keep detect “sudden” changes

Sampling frequency

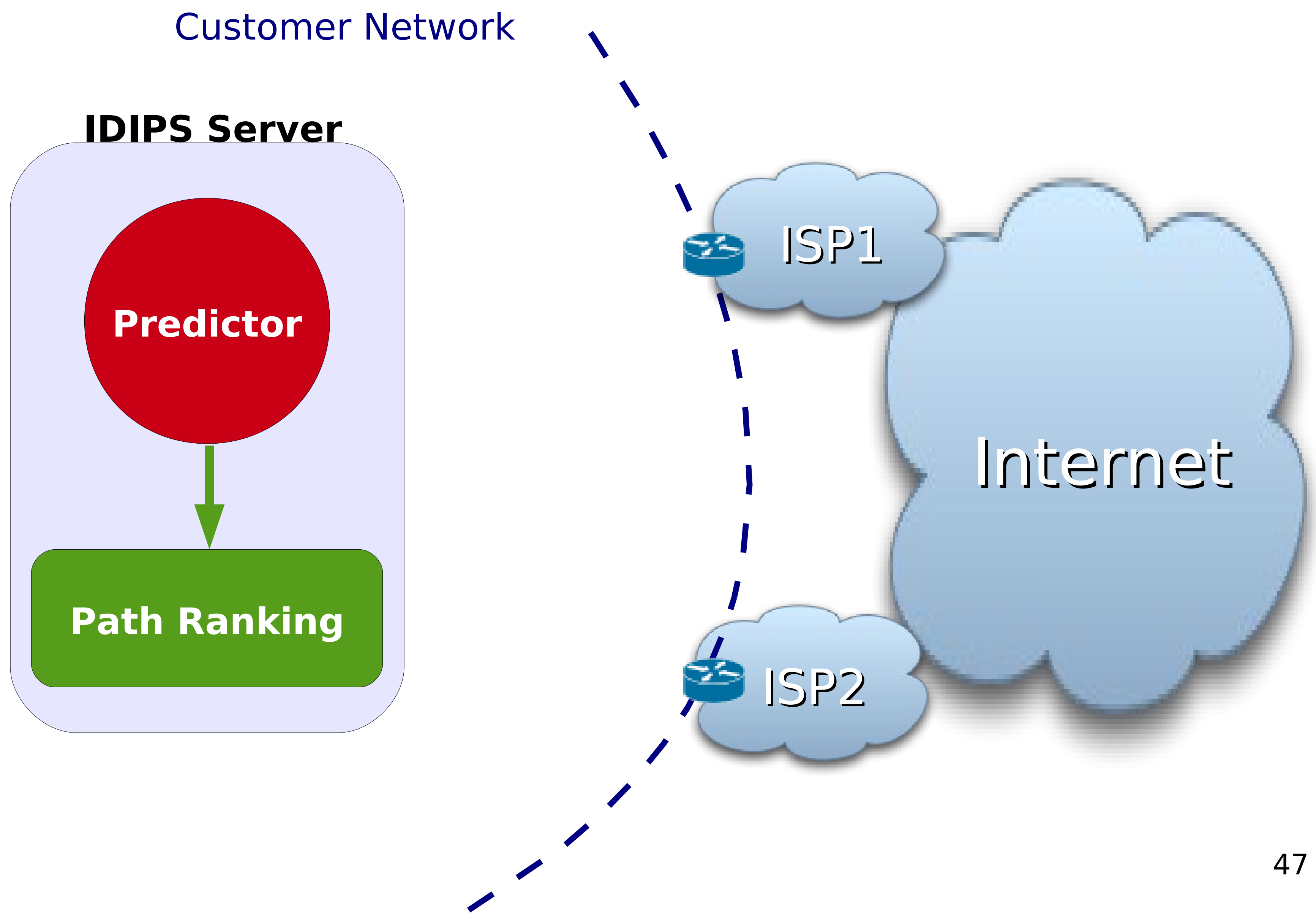


Path Performance Prediction



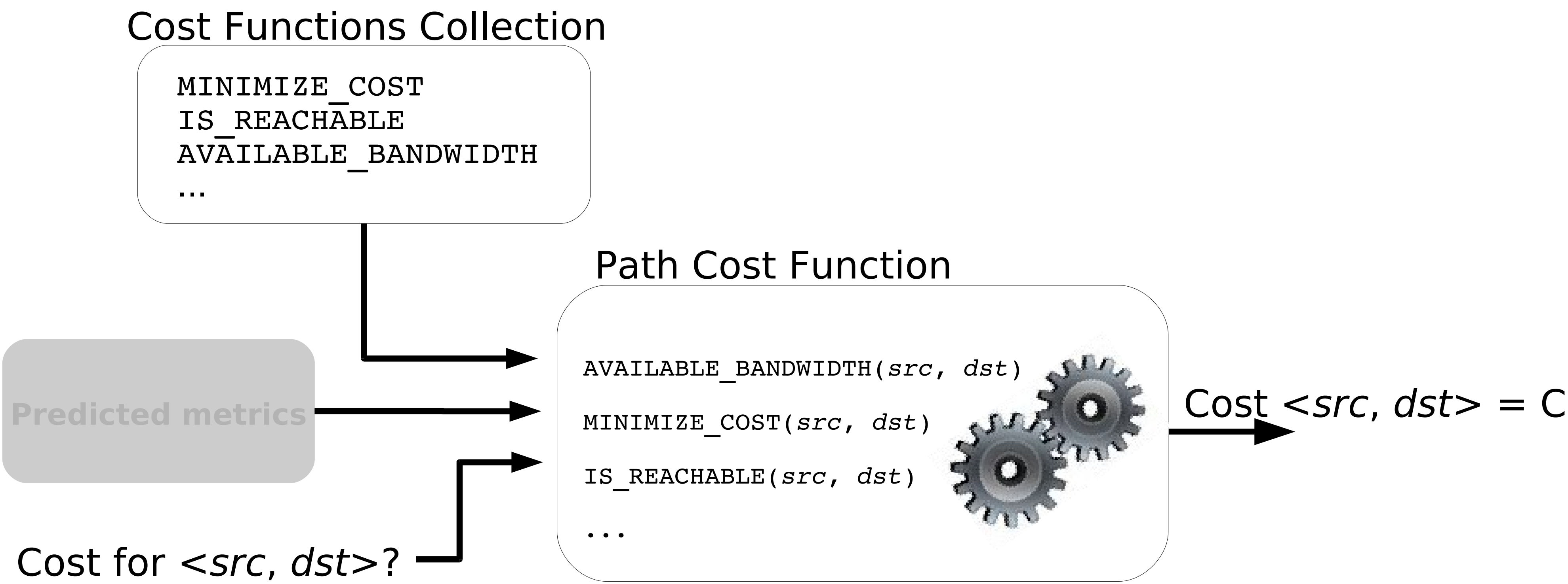
Inside IDIPS
(Path Ranking)

Inside IDIPS



Path Ranking

- Compute a cost for each path (on-demand)

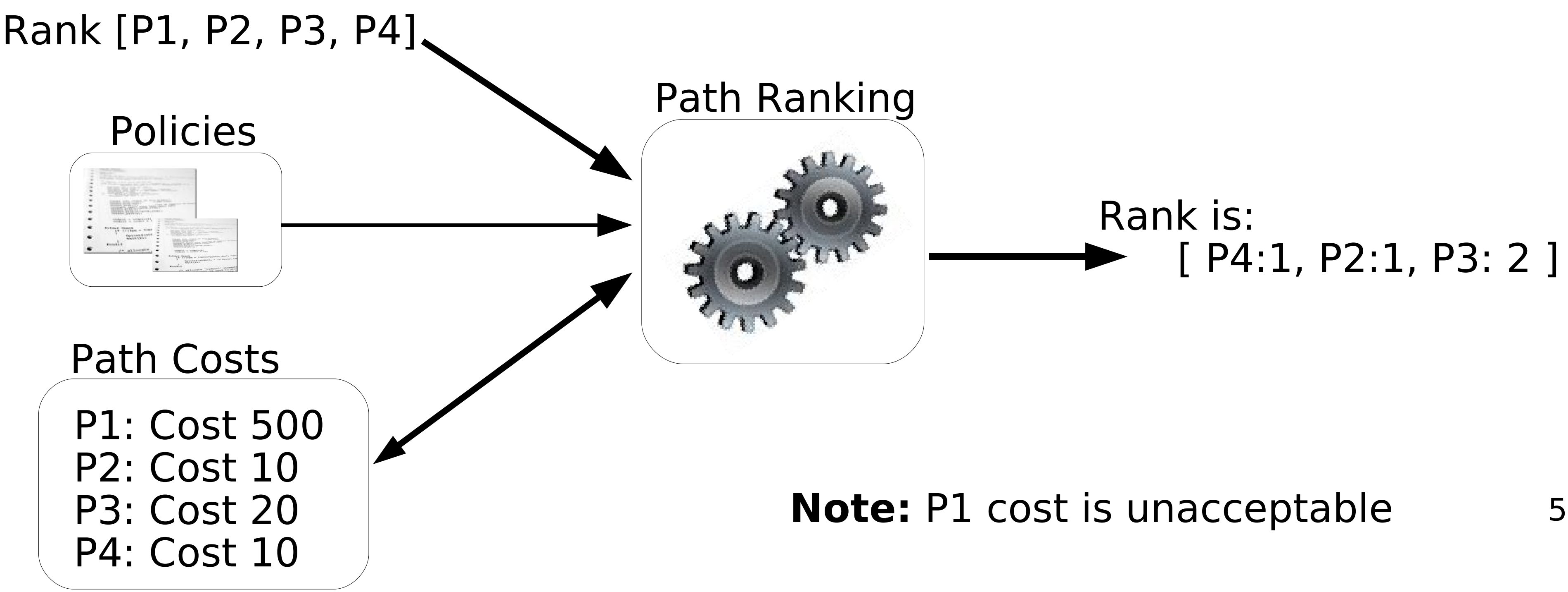


Cost Function

- A cost function gives the cost of a path regarding a given (set of) metric(s)
- Parameter
  - A path, described as a <*src*, *dst*> pair
- Returned value
  - An integer representing the cost
- *Transitivity* with cost function relationship
- The *lowest* the cost, the *better* the path

Rank the paths

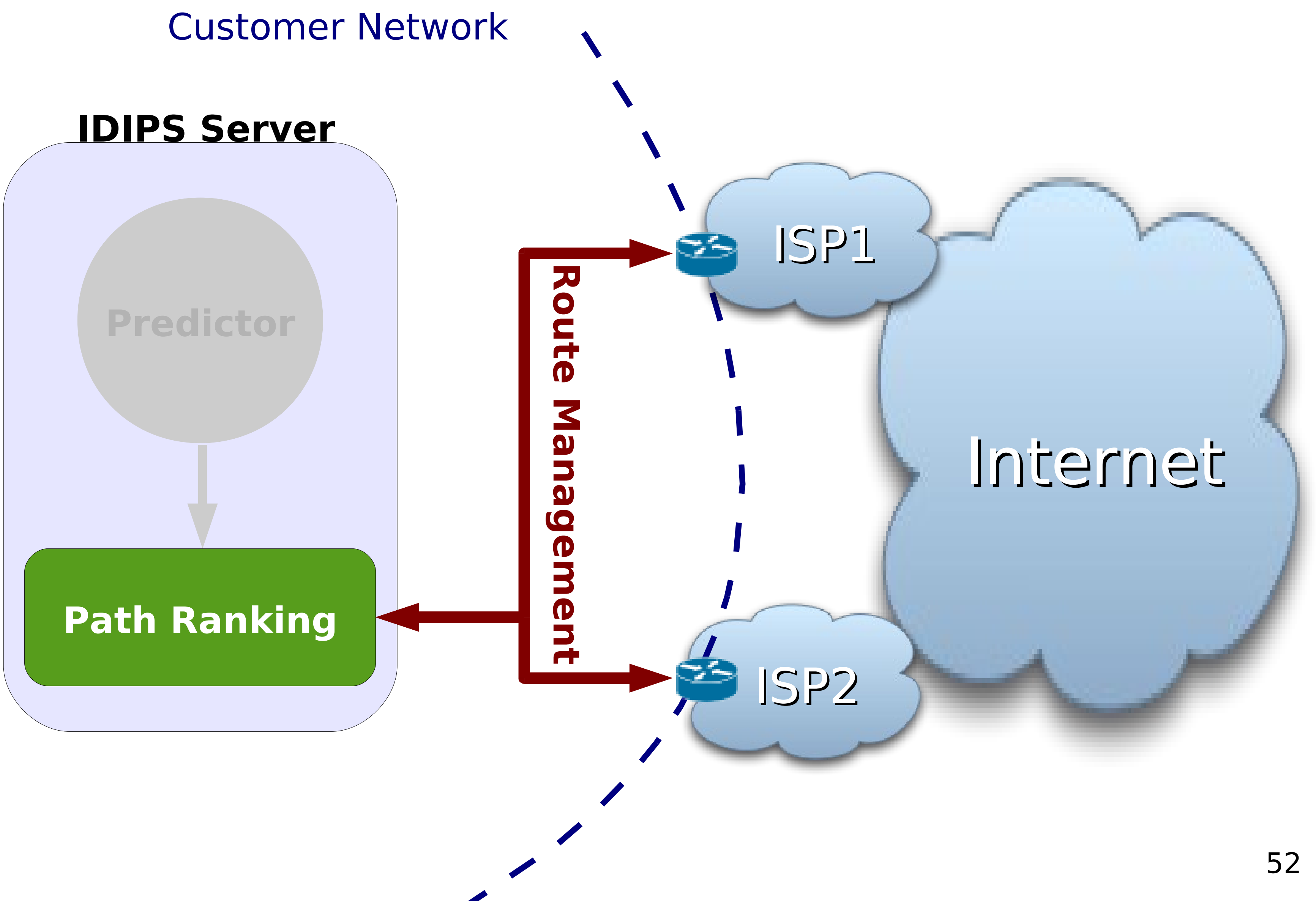
- Rank is an abstraction of cost
  - The smaller, the better
  - Cost is absolute, rank is relative
  - Cost relationship is transitive, not the ranking



Note: P1 cost is unacceptable

Inside IDIPS
(Route Management)

Inside IDIPS



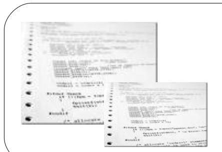
Route Management

BGP Routing Information Base

Network	Next Hop	LocPrf	Path
>A/a	R2	100	AS5
…			
>P/p	R1	2000	AS6:AS3
P/p	R2	100	AS6:AS3:AS3:AS1
P/p	R3	2000	AS6:AS4
…			

Path Ranking

Policies



Decision Engine

Routing Engine

Network	Next Hop	Rank
>A/a	R2	1
…		
P/p	R1	2
P/p	R2	3
>P/p	R3	1
…		

Forwarding Engine

Network	Interface
A/a	interface2
P/p	interface3

Conclusion

Conclusion

- Today's interdomain traffic engineering
  - Art
  - Mostly ignore path performances
- Informed Path Selection is required
  - Control the costs
  - Improve performance
  - Simplify management

Further Works

- Can we combine different metrics to have a better prediction?
- Can we predict several metrics from other ones (e.g., bandwidth from delay)?
- How to decentralize the ranking and keep route management coherent?
- How to predict sudden changes?

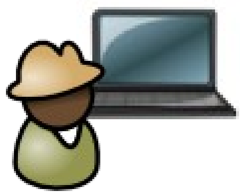
Thank you

?? || /**/

Backup Slides

Internet Today
(Seen by the users)


YAHOO!
LinkedIn
The New York Times
Expect the World
NYSE
amazon.com

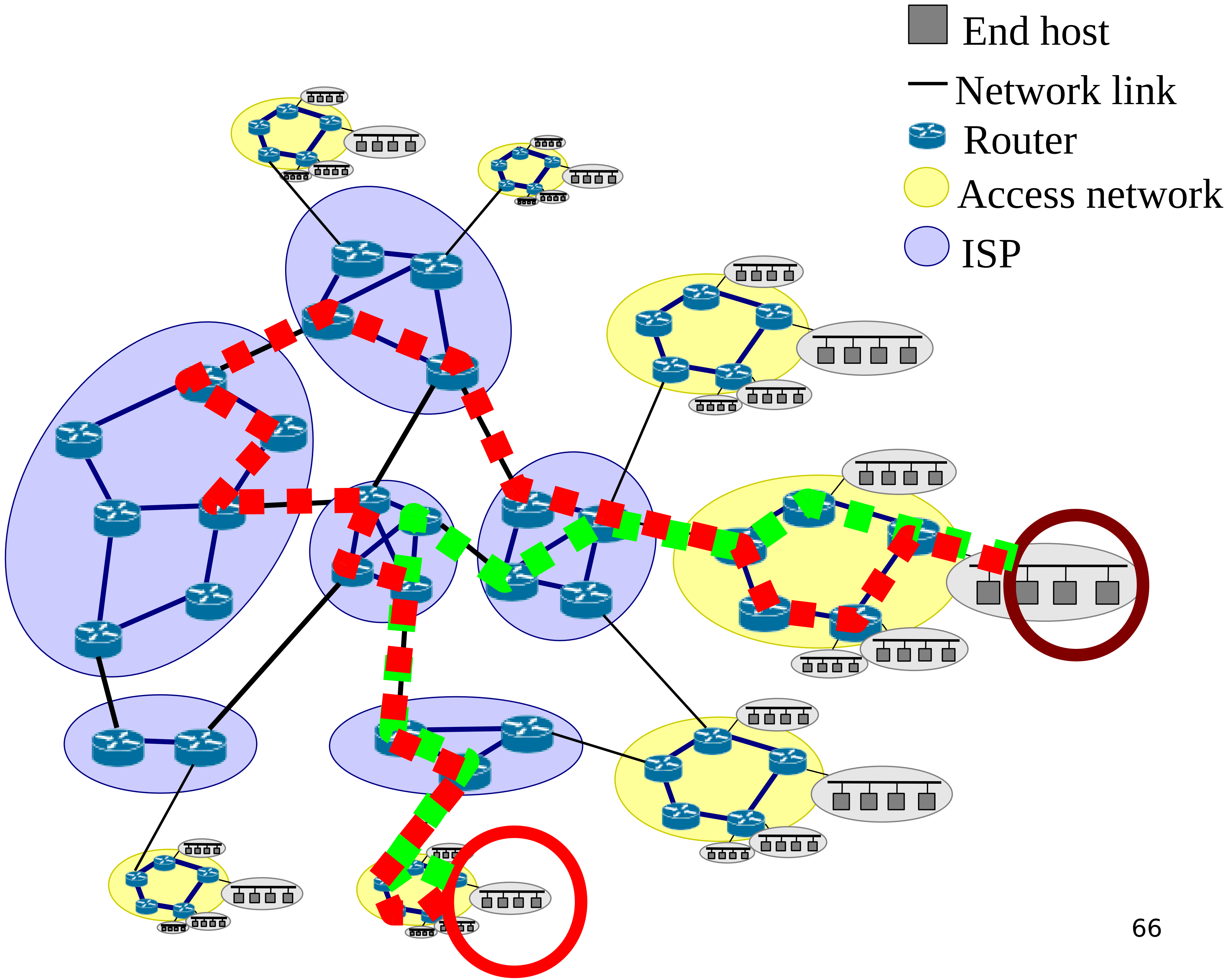



YAHOO!
The Pirate Bay
Google
SOURCEFORGE
.net
BitTorrent


Internet

Internet Today

(What is hidden by the  ?)



Inspired by http://interstices.info/jcms/c_15921/internet-le-conglomerat-des-reseaux

How to select the best path?

Internet Today
(The basis of Interdomain routing)

Interdomain routing

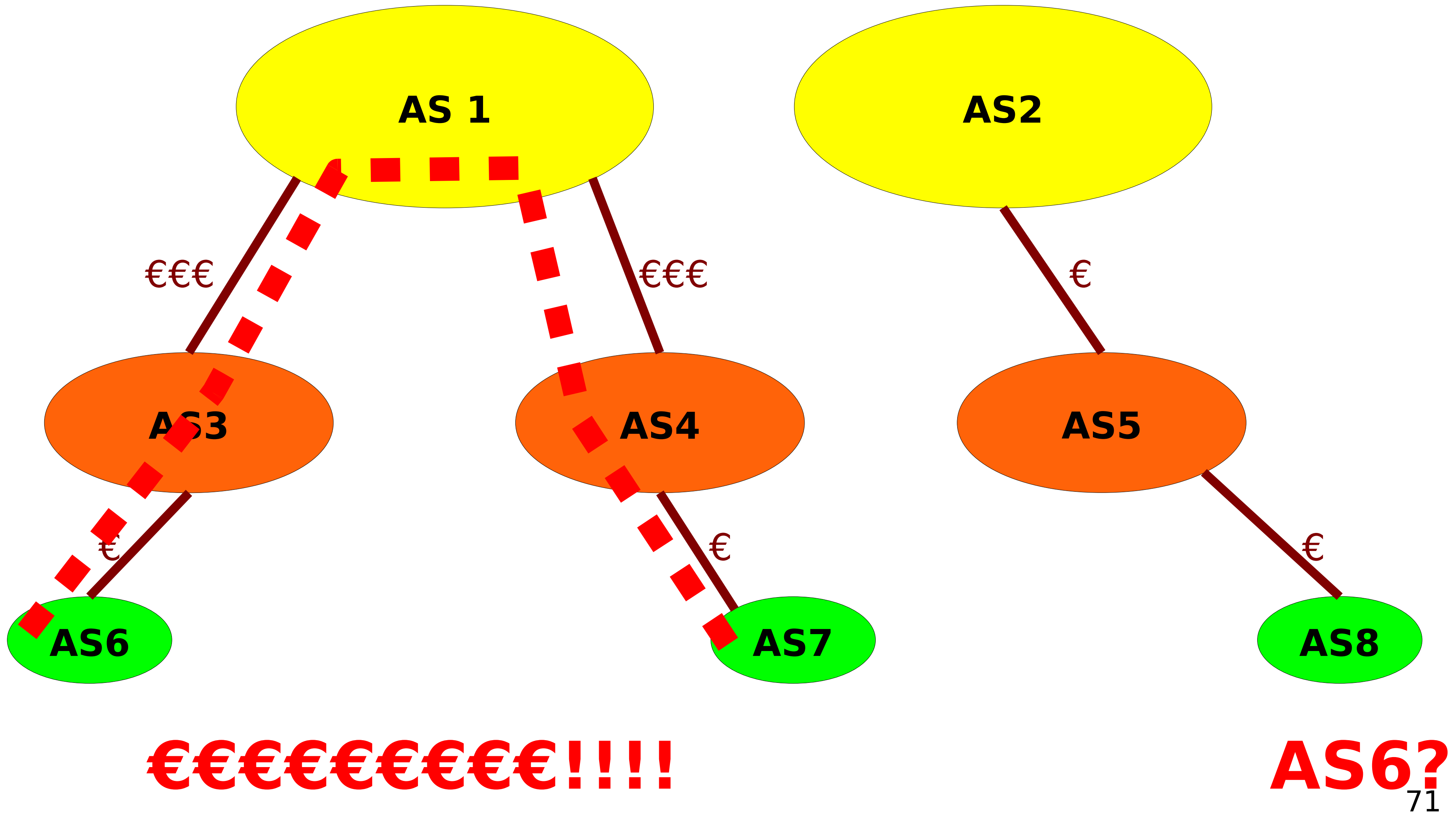
- Goal
  - Allow to transmit data along the best path towards the destination through several *transit domains* while taking into account the *routing policies* of each domain without knowing the detailed topology of those domains

- The *Border Gateway Protocol* (BGP) is the common protocol between the domains

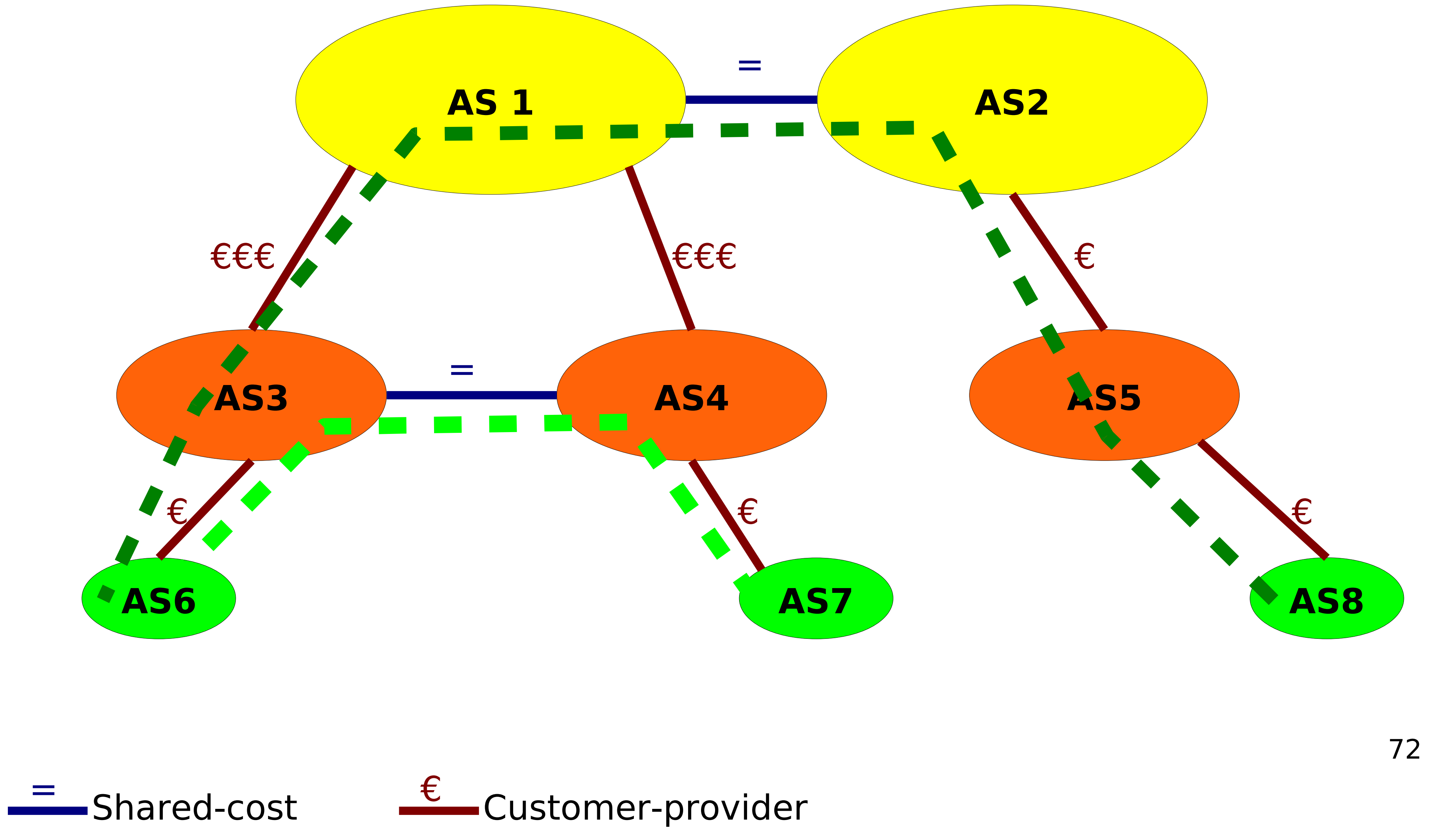
Routing policies

- In theory, BGP allows each domain to defines its own routing policy…
- In practice, there are two common policies:
  - ***Customer-provider* peering:** customer *c* buy Internet connectivity to provider *p.*
  - ***Shared-cost* peering:** domains *x* and *y* agree to exchange data by using a direct link through an interconnection point.

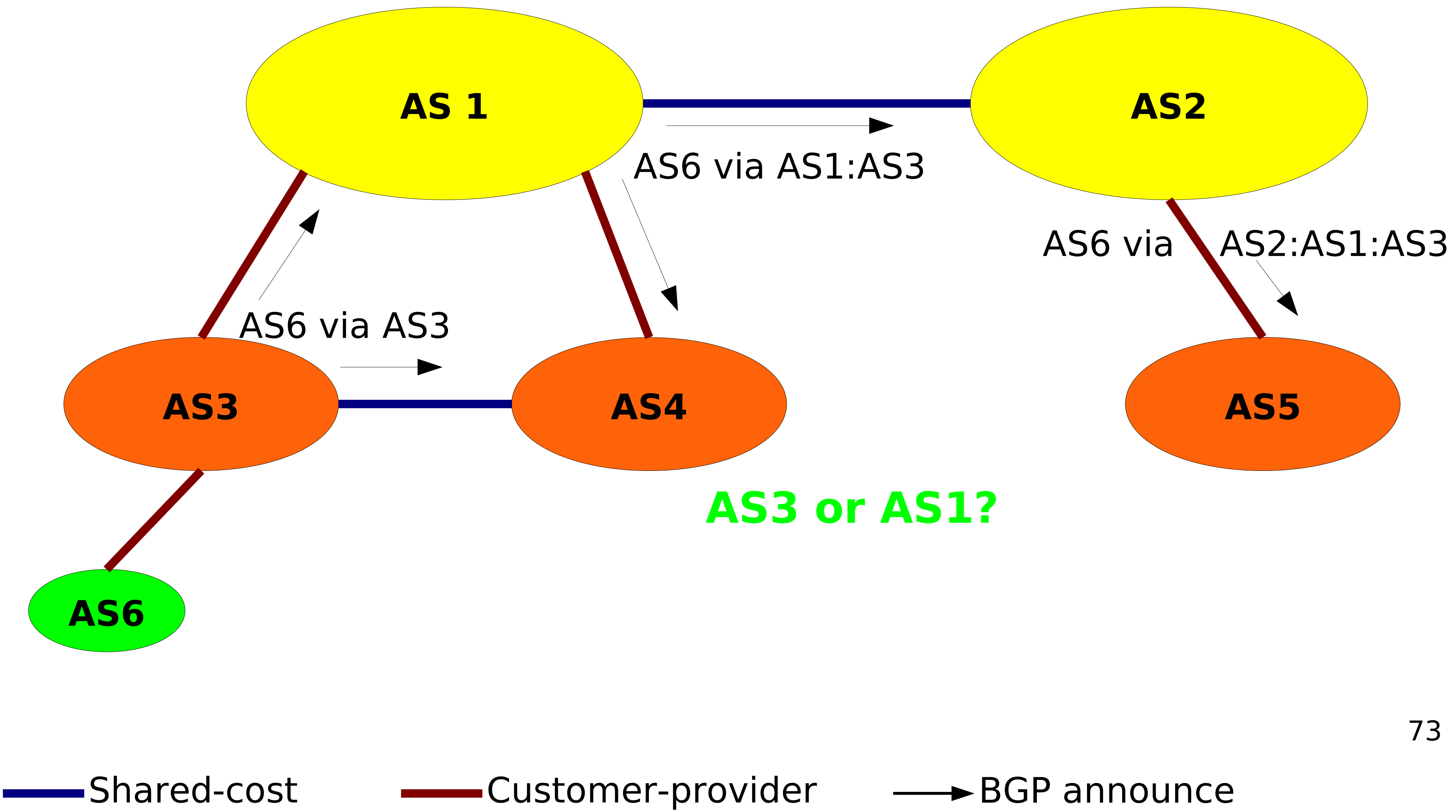
Customer-provider peering



Shared-cost peering



How routes are discovered?

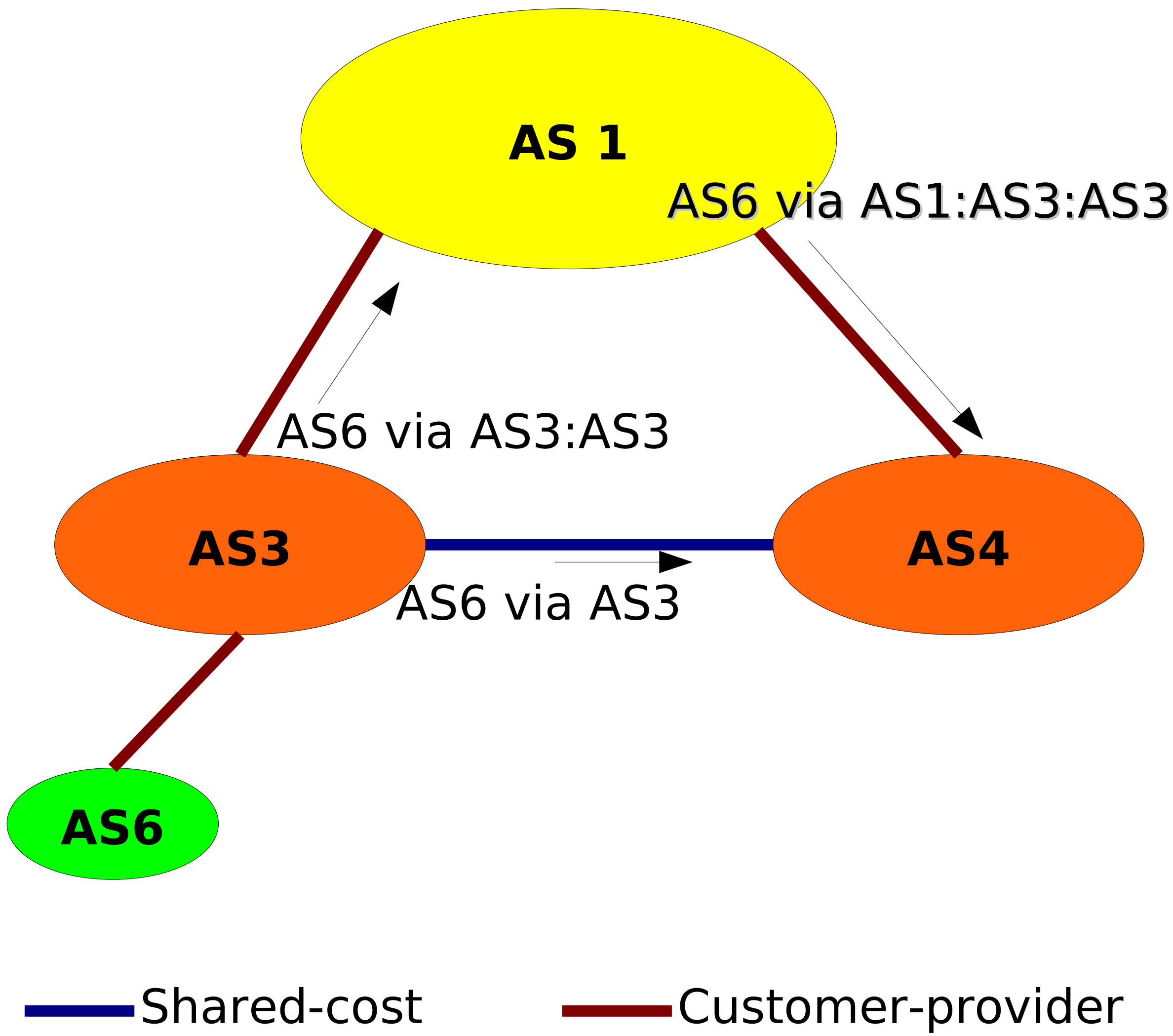


Simplified BGP decision process

1. Select routes with the highest `local-pref`
   - Manual configuration
2. If there are several routes, chose routes with the shortest AS path
   - Mostly determined by the topology
   - Can be influenced by using pre-pending
3. If there are still routes tie-breaking rule

Route control with BGP

- Manual configuration!



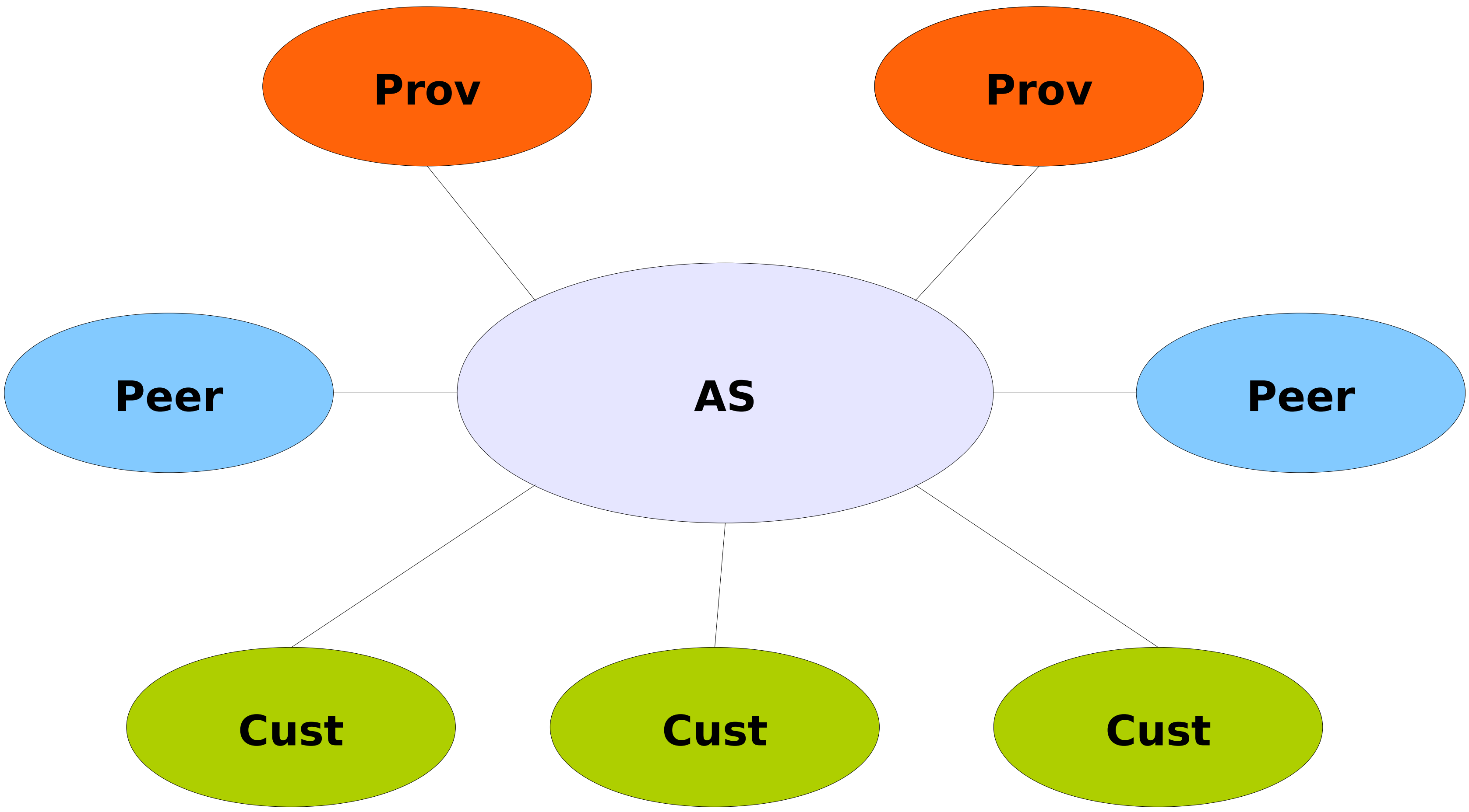
Policy for AS3:

```
Export:
    To AS1 set as-path
                  prepend AS3
```

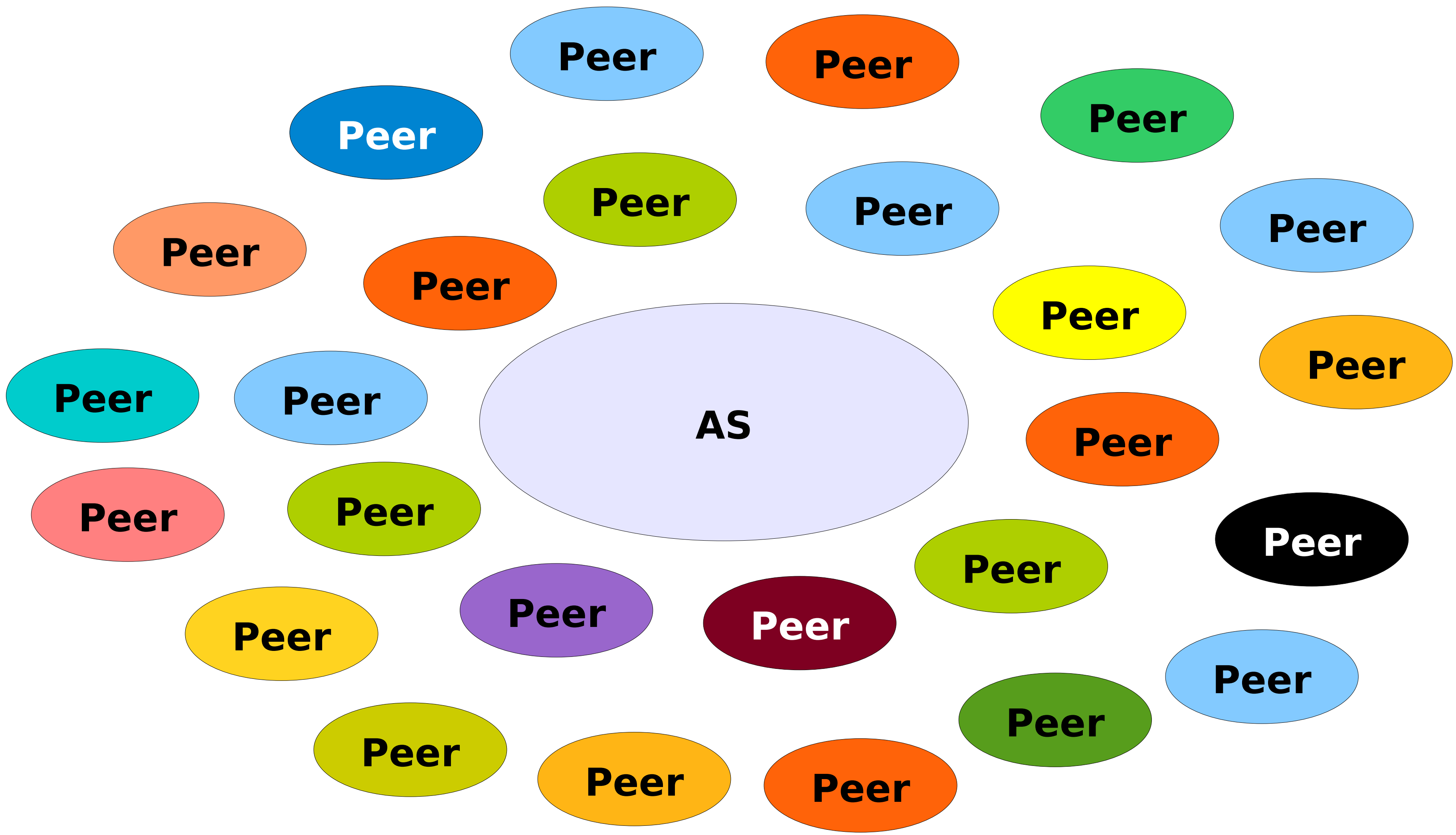
Policy for AS4:

```
Import:
    From AS3 set localpref=2000
    From AS1 set localpref=100
```

The simple case ...



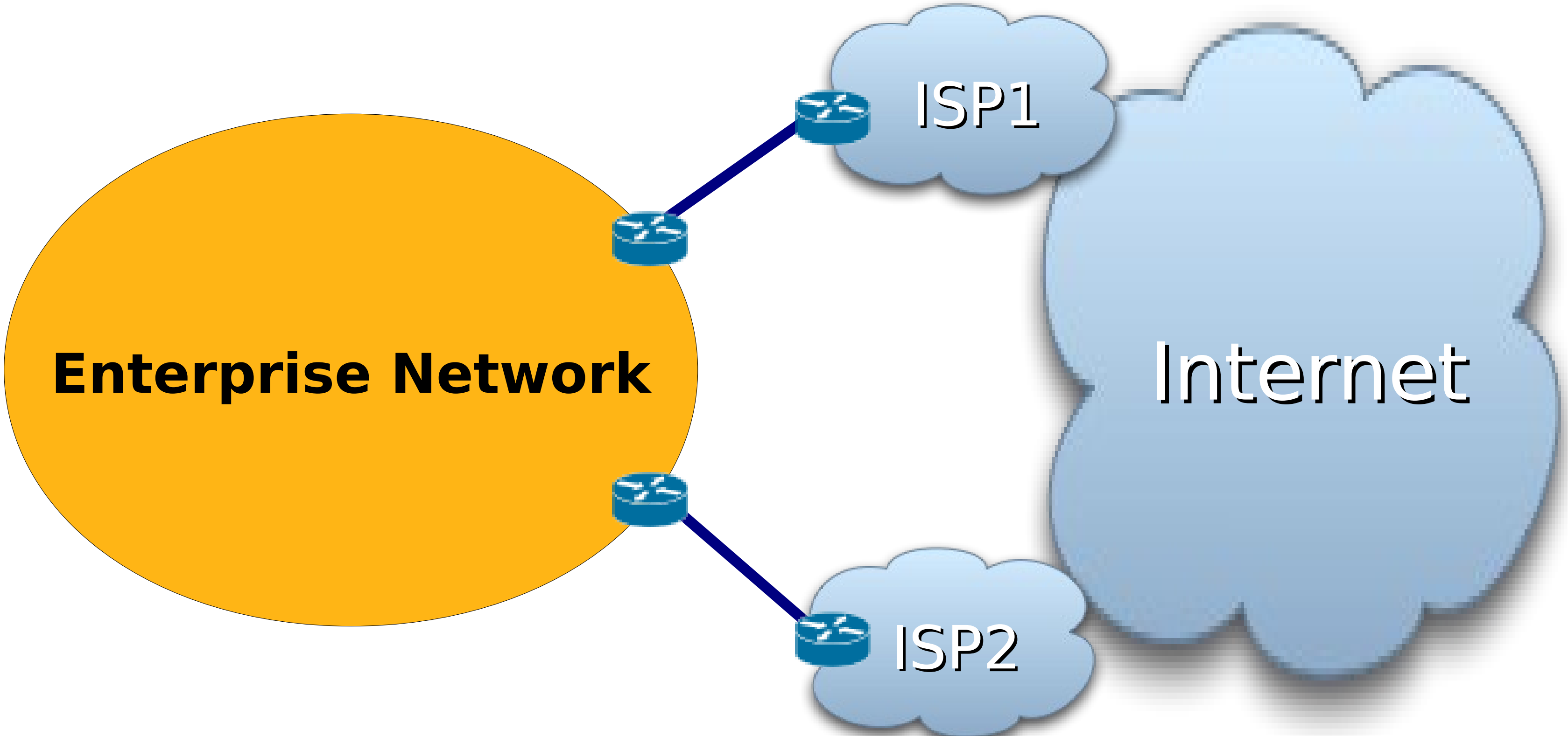
... the nightmare



What is missing?

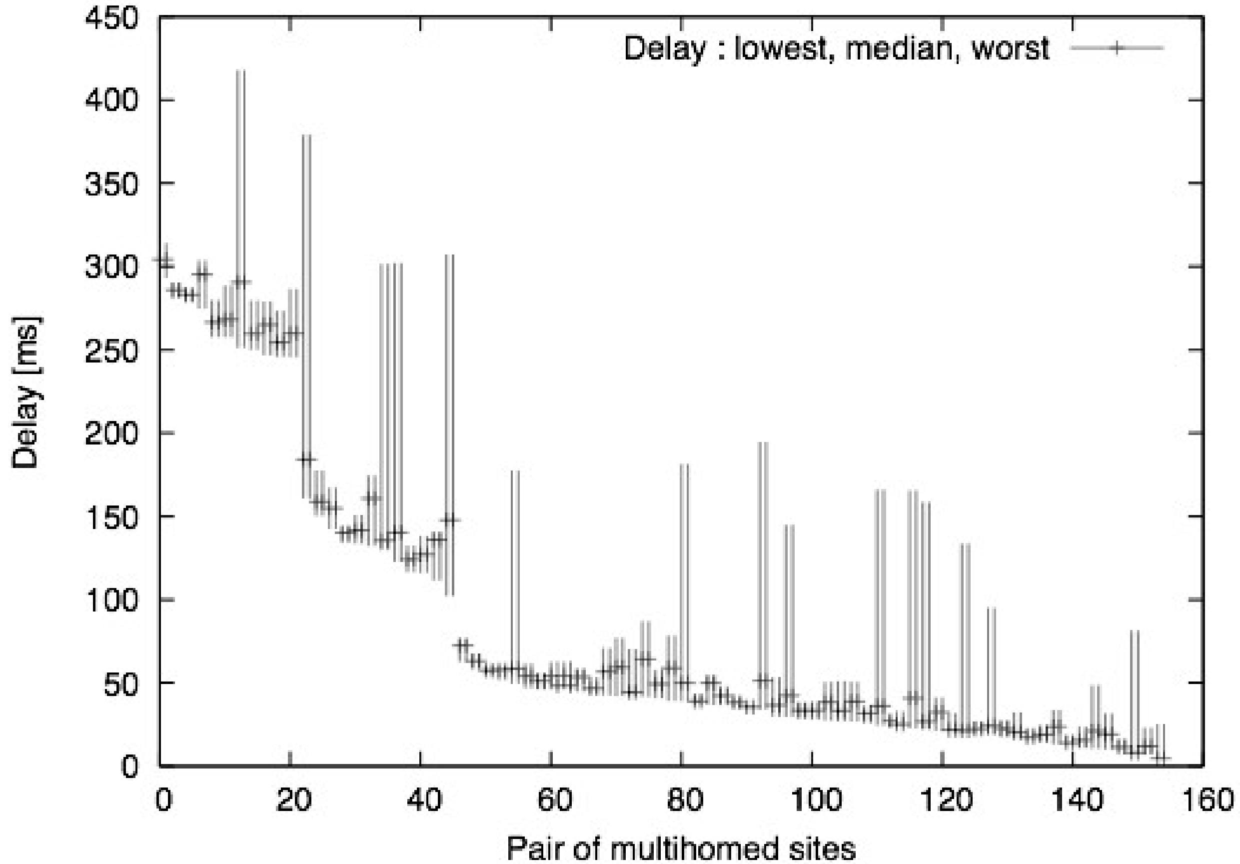
Path are not equal

- Today: the cheapest
- but... multi-homing is common



Path are not equal

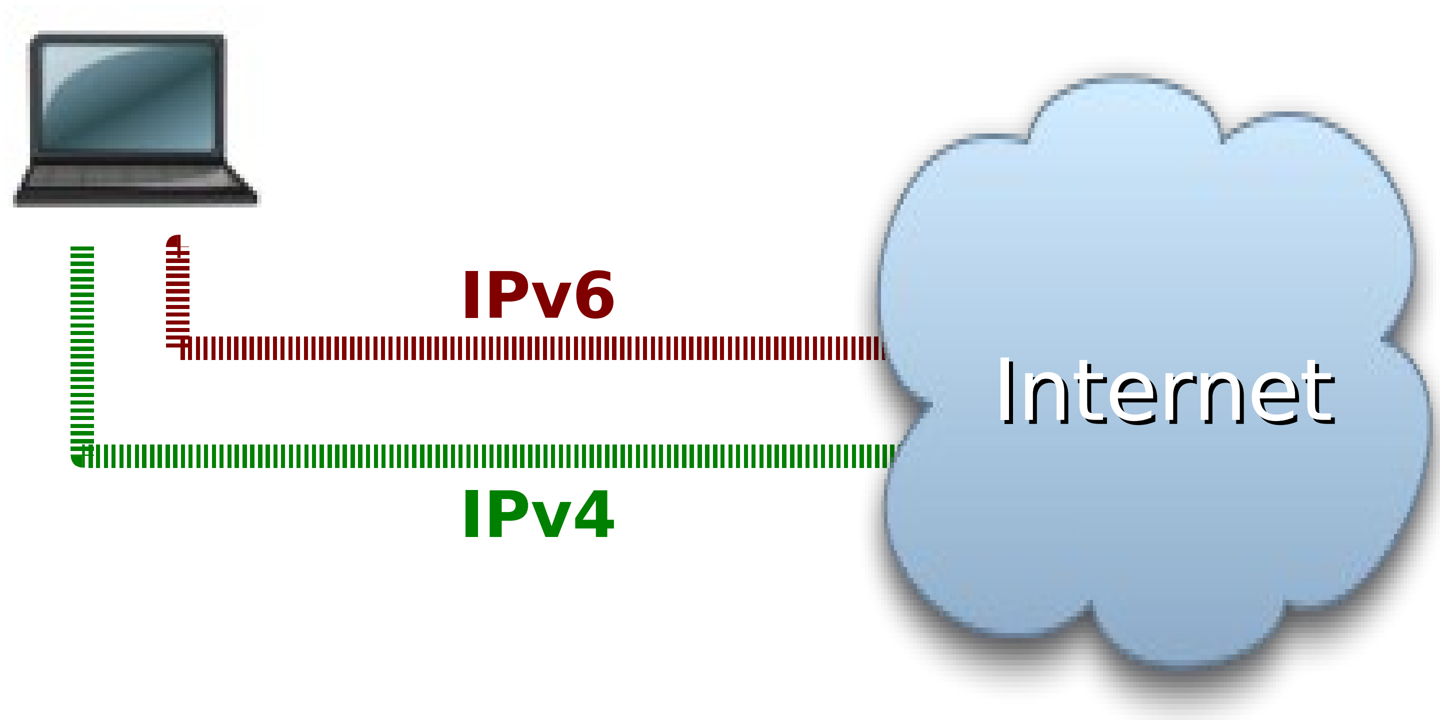
- Today: the cheapest
- but... multi-homing is common



[QILB07] B. Quoitin et al., Evaluating the Benefits of the Locator/Identifier Separation, MobiArch 2007

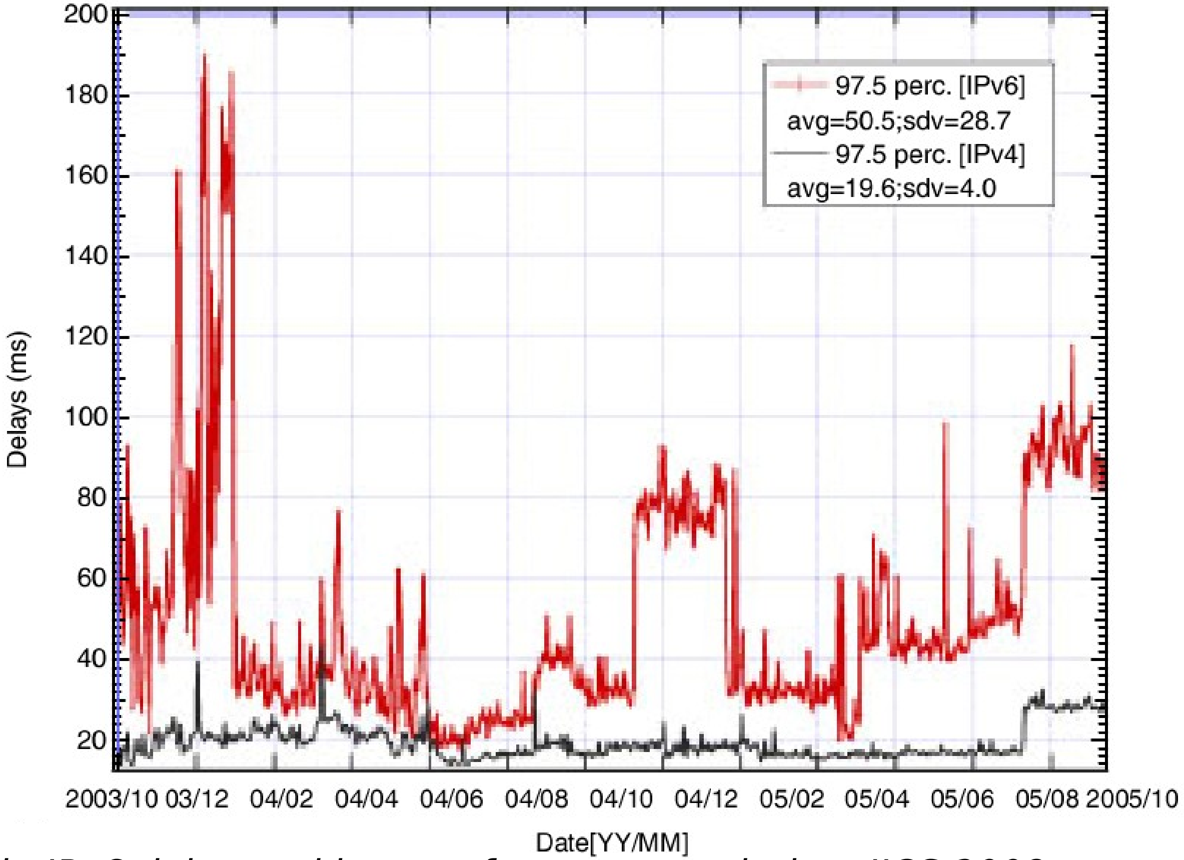
Path are not equal

- Today: the cheapest
- but... multi-protocol stacks are arriving



Path are not equal

- Today: the cheapest
- but... multi-protocol stacks are arriving



[ZJUM08] X. Zhou et al., IPv6 delay and loss performance evolution, IJCS 2008

Path are not equal

- Today: the cheapest
- but... applications/services need QoS

Delay
Jitter
Bandwidth
Loss

The best path?

- Today: the cheapest
- Tomorrow: the more adapted
  - The cheapest
  - The faster
  - The safer
  - The more stable
  - ...

Cost Function Example

- Always maximize the bandwidth for premium users
- Always minimize the cost for standard users
- Maximize the bandwidth during the night for advanced users but minimize the cost during the day

Define the building blocks

Algorithm 2 Example of Cost Function for the cost minimization

Ensure: Integer value representing the cost of using the path defined by *src*, *dst*.

```
procedure MINIMIZE_COST_CF(src, dst)
    attributes ← path_attributes(src, dst)
    return attributes{'COST'}
end procedure
```

Algorithm 3 Example of available bandwidth Cost Function

Ensure: Integer value representing the result of this Cost Function.

```
procedure AVAILABLE_BW_CF(src, dst)
    attributes ← path_attributes(src, dst)
    return (MAX_BW – attributes{'ABW'})
end procedure
```

The highest the bandwidth, the better

Algorithm 4 Example of customer family Cost Function

Ensure: Integer value representing the customer family for traffic from *src* to *dst*.

```
procedure CUSTOMER_FAMILY_CF(src, dst)
    attributes ← path_attributes(src, dst)
    return attributes{'FAMILY'}
end procedure
```

Premium: 1
Advanced: 10
Standard: 20

Combine the building blocks

Algorithm 5 Example of customer family Cost Function

Ensure: Encounters customers requirements

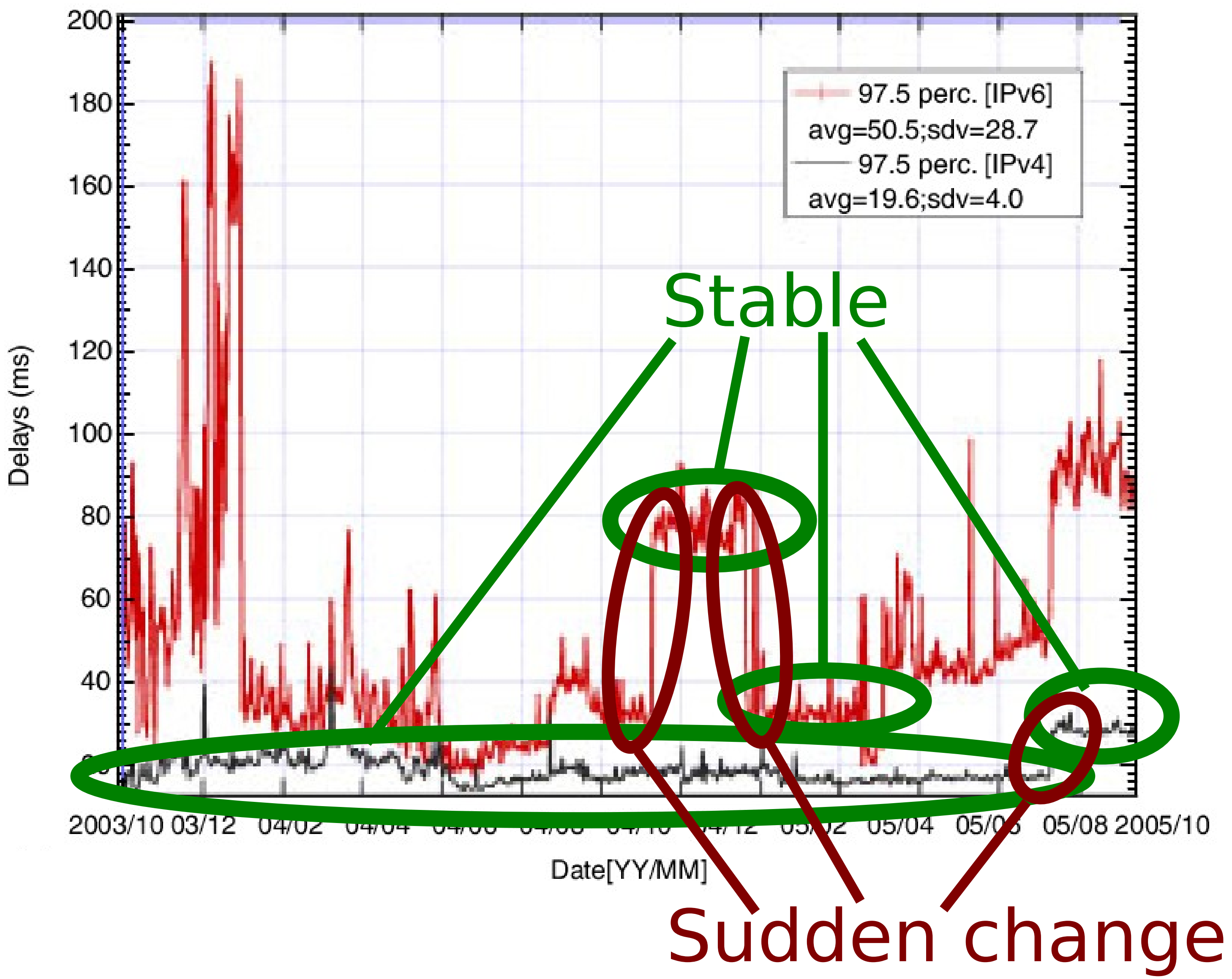
```
procedure CUSTOMER_MANAGEMENT_CF(src, dst)
    if (is_reachable_cf (src, dst) = 2) then
        return (UNREACHABLE)
    end if
    customer ← CUSTOMER_FAMILY_CF(src, dst)
    if (customer == 1) then
        return (AVAILABLE_BW_CF(src, dst))
    end if
    if ((customer == 10 ∧ DAY) ∨ (customer = 20)) then
        return (MINIMIZE_COST_CF(src, dst))
    end if
    if (customer == 10 ∧ NIGHT) then
        return (AVAILABLE_BW_CF(src, dst))
    end if
    return (ERROR)
end procedure
```

Premium user

Advanced user

Standard user

How to react/detect to sudden changes?



[ZJUM08] X. Zhou et al., IPv6 delay and loss performance evolution, IJCS 2008