An ALTO Protocol: P4P/Info Export

Richard Alimi, Doug Pasko, Reinaldo Penno,

Laird Popkin, Satish Raghunath, Stanislav Shalunov,

Yu-Shun Wang, Richard Woundy, Y. Richard Yang

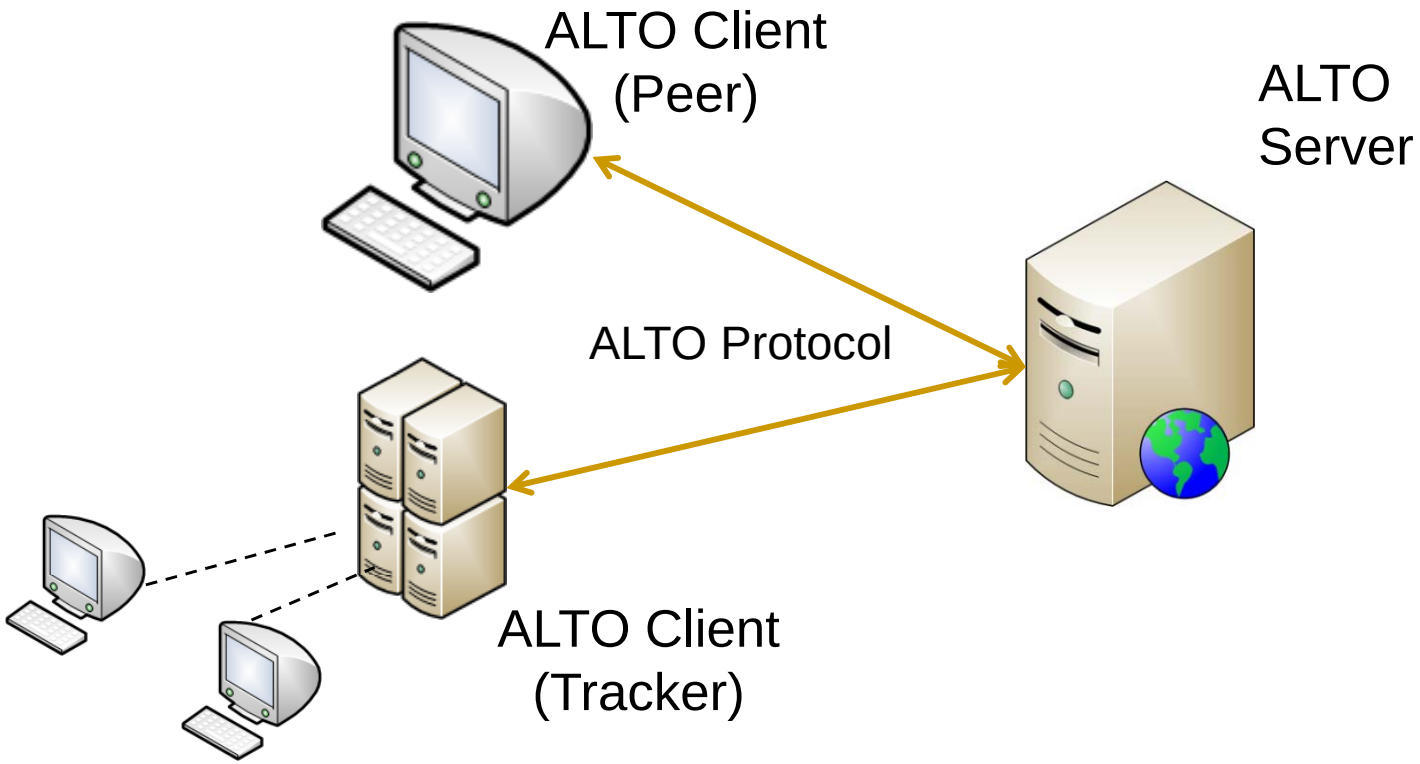
Presenters: Y. Richard Yang, Reinaldo Penno

Outline

- Key architectural concepts
- Protocol specification
- Comparison with other schemes

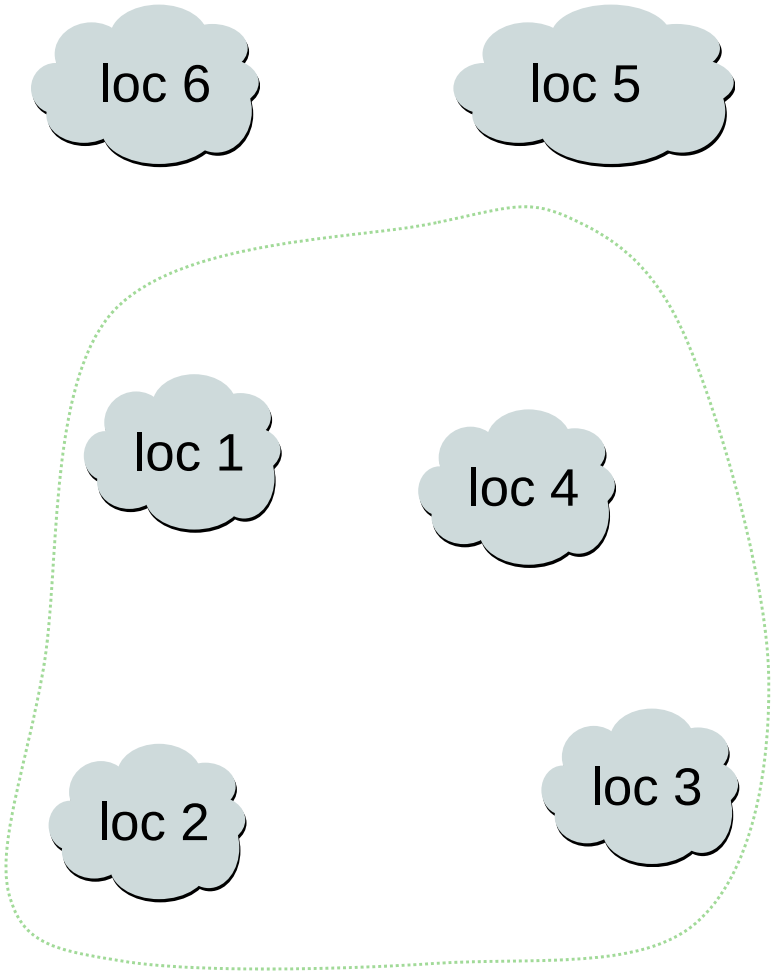
Objective

- Provides network information to P2P applications to achieve better peer selection.



My-Internet View

- Each ALTO Server maintains a "my-Internet" view
- A my-Internet view consists of
  - A set of network locations;
  - ALTO Cost between each pair of network locations for a given ALTO Cost Type.

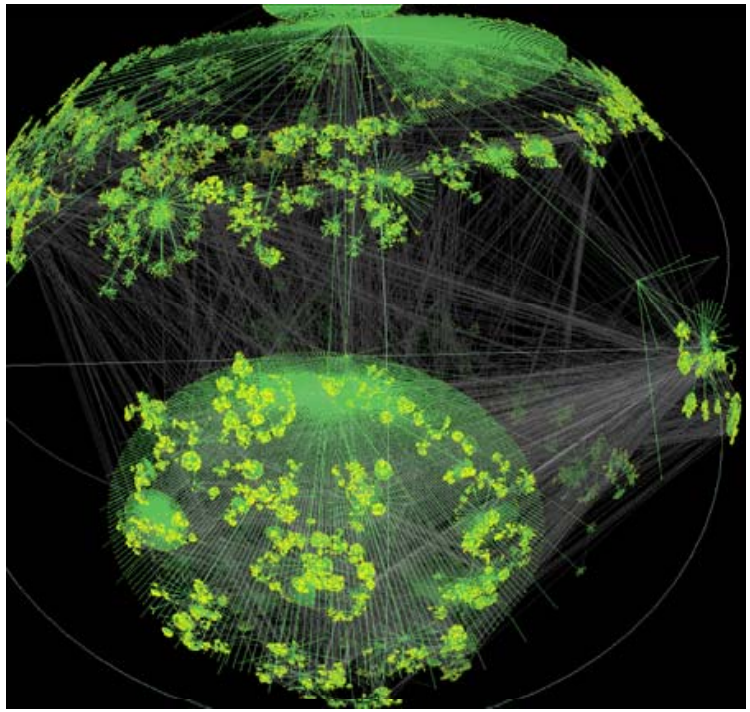


ALTO Query/Response

- Basic ALTO Query:
  - A set of source (resource consumer) network locations on the “my-Internet” view
  - A set of destination (resource provider) network locations on the “my-Internet” view
  - An ALTO Cost Type
  - Mode: numerical or ordinal (ranking)
- ALTO Response:
  - Numerical values/ranking from given sources to given destinations

A Key Architectural Issue

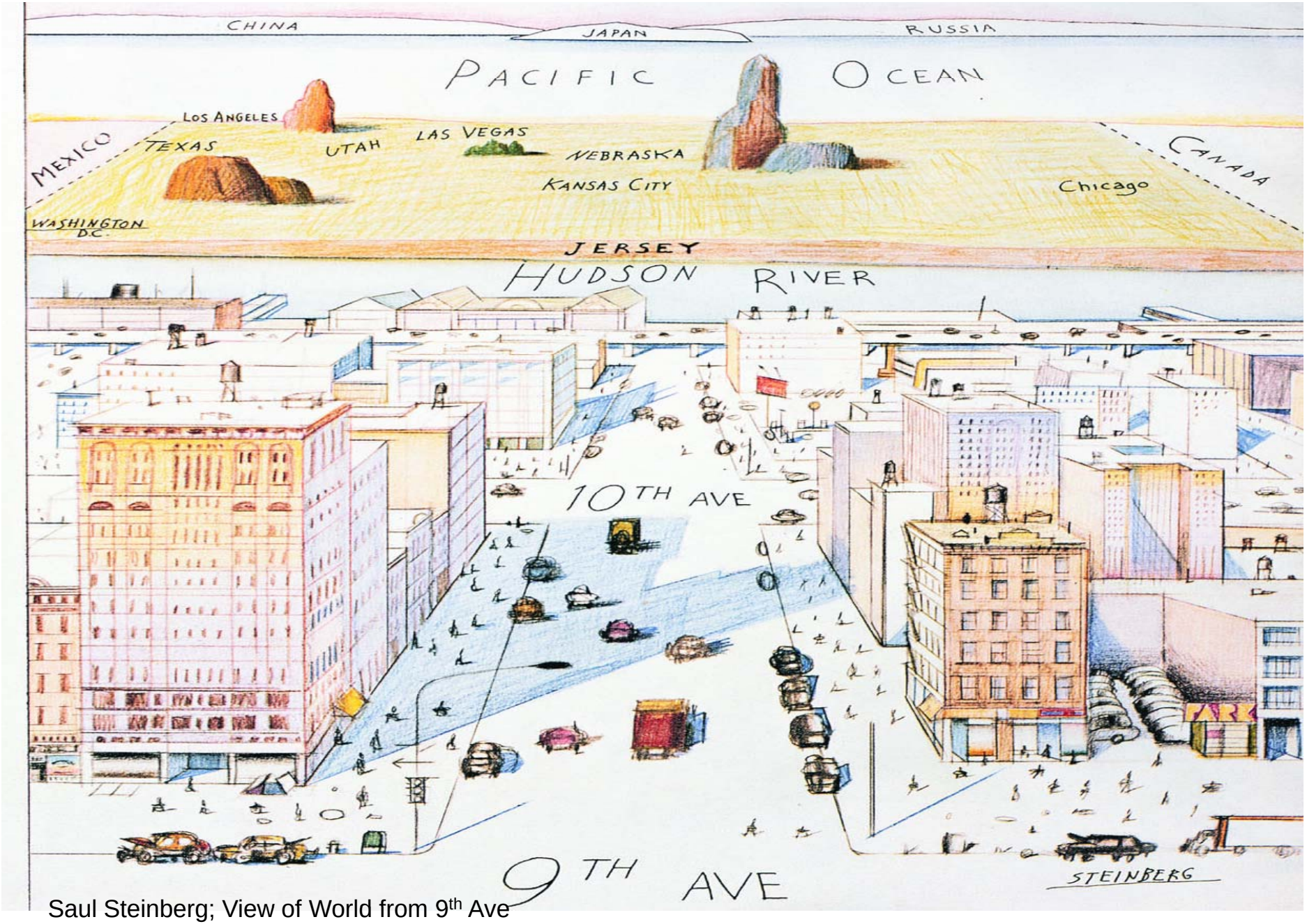
- How to specify the sets of source and destination network locations?
  - Important for scalability, privacy, & service availability



Representation of the structure of the Internet. Young Hyun of CAIDA, using graph visualization tool code-named Walrus

Source/Destination Grouping

- A Source Group is a set of network locations that have similar ALTO Costs to other network locations
- A Destination Group is a set of network locations that have similar ALTO Costs from other network locations
- A Source/Destination Group may represent an IP Prefix, a point of presence (PoP), a type of customers (wireless, DSL, FiOS), an AS, or a set of AS(es)



Saul Steinberg; View of World from 9th Ave

Van Jacobson: “you want to use a hierarchical identifier space so that you can aggregate out detail as you get far away.”

Grouping Benefits

- Grouping improves scalability
  - Source Group enables *caching & redistribution*
  - Destination Group enables compact representation of ALTO info
  - Reduces load/overhead on ALTO Server and P2P Trackers
  - Allows group & cost info to update at different timescales
- Grouping helps privacy
  - Peer perspective – masquerading individual peers
  - ISP perspective – avoid revealing fine-grained network topology by probing/learning experiments

Outline

- Key architectural concepts
- ALTO Protocol Design
- Comparison with Other Proposals

ALTO Protocol: Key Design Features

- Clean instantiation of proposed architecture
- Maps well into existing P4P/Info Export implementations
- Transport – based on HTTP
  - Availability & maturity of HTTP clients & servers
  - Scalability – Can utilize HTTP caching
  - Flexible ALTO server implementation strategies
- Decoupling message format from HTTP
  - Evolve independently
  - Make it possible to use P2P redistribution
  - Adaptable to other transports
- Textual encoding (currently based on SDP format)
  - Ease understanding and debugging
  - leverage existing SDP implementation

ALTO Network Information: Interfaces

- `GetNetworkMap/GetNetworkIdentifier:` Mapping between network locations and Source/Destination Groups

- `GetCostMap:` Defines costs amongst network locations and Source/Destination Groups

Interface Descriptor

- An ALTO Response may specify a URI:

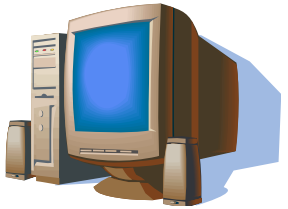
```
GetNetworkMap-Complete    /alto/networkmap-complete.txt
GetNetworkMap-Source      /alto/networkmap-<SRC-GRP>.txt
GetNetworkMap             /alto/msg.cgi?getnetworkmap

GetCostMap-Complete       /alto/costmap-complete.txt
GetCostMap-Source         /alto/costmap-<SRC-GRP>.txt
GetCost                   /alto/msg.cgi?getcost
```

- Benefit: Allows flexible ALTO Server implementation strategies (e.g., pre-generated files, DB backend, .torrent, etc)

Example 1: ALTO Client Embedded in App. Tracker

ALTO Server
(alto.isp.net:80)

ALTO Client
@appTracker



1. Request Interface Descriptor

```
i=getcost                          /alto/msg.cgi?q=cost
i=getcostmap-complete              /alto/cost-complete.txt
i=getnetworkmap                    /alto/msg.cgi?q=netmap
i=getnetworkmap-complete           /alto/netmap-complete.txt
…
```

2. Retrieve Network Map (cacheable)

```
GET /alto/netmap-complete.txt
```

```
r=getnetworkmap
m=PID1 128.36.0.0/16
m=PID2 130.132.0.0/16
m=PID3 65.198.30.0/24
```

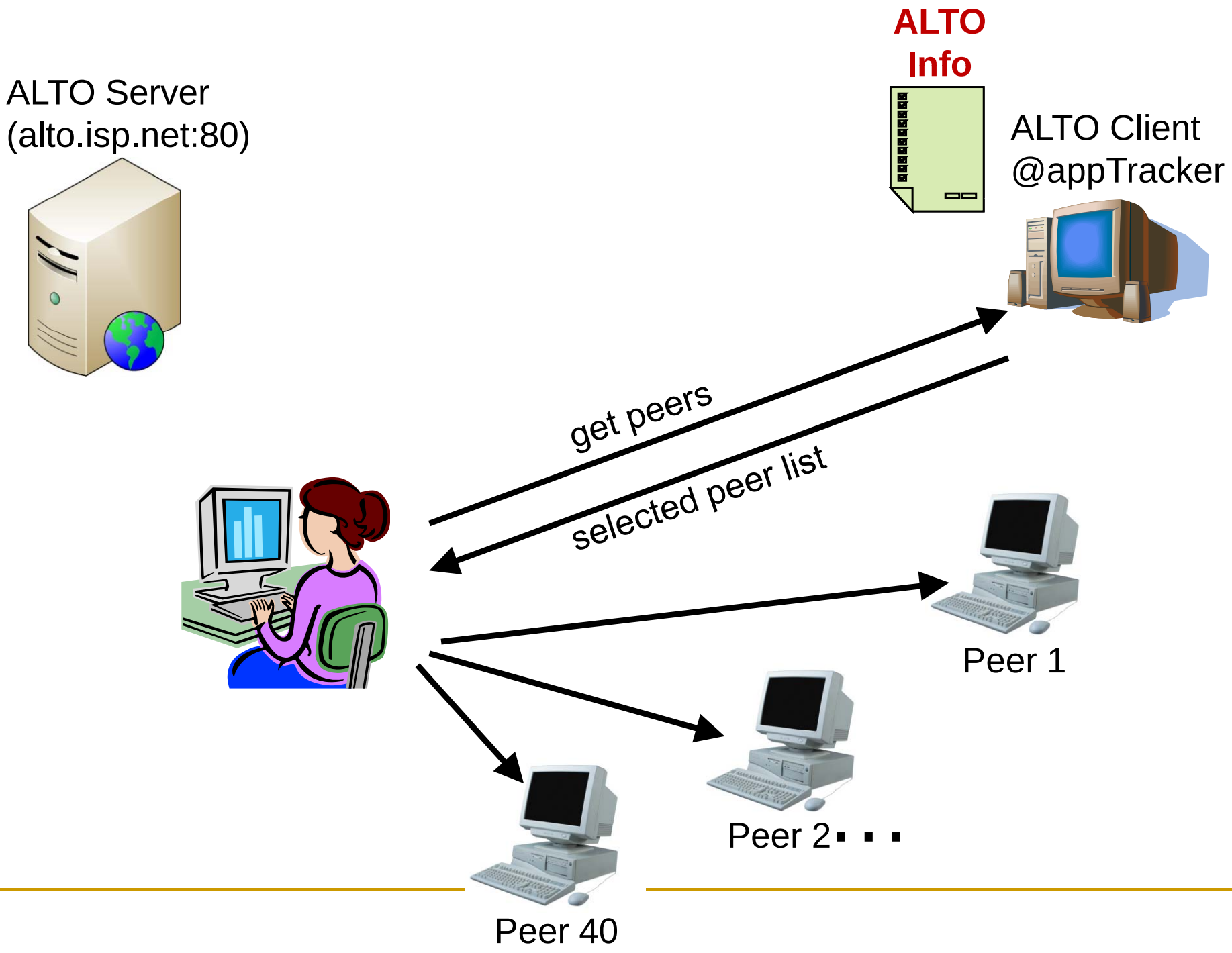
netmap-complete.txt can be cached at HTTP caches

3. Retrieve Cost Map (cacheable):

```
GET /alto/costmap-complete.txt
```

```
r=getcost COST routing numerical
c=SRC PID1
DEST PID1 1
DEST PID2 8
DEST PID3 3
c=SRC PID2
…
```

costmap-complete.txt can be cached at HTTP caches


ALTO
Info
ALTO Server
(alto.isp.net:80)
ALTO Client
@appTracker
get peers
selected peer list
Peer 1
Peer 2
Peer 40

Example 2: ALTO Client Embedded in a P2P Client

ALTO Server
(alto.isp.net:80)

ALTO Client
@P2P Client
(in PID1)

1. Request Interface Descriptor

```
i=getcostmap-source        /alto/costmap-PID1.txt
i=getnetworkmap-source     /alto/netmap-PID1.txt
p=PID1
…
```

2. Retrieve Network Map (cacheable)

```
GET /alto/netmap-PID1.txt
```

```
r=getnetworkmap
m=PID1 128.36.0.0/16
m=PID2 130.132.0.0/16
m=PID3 65.198.30.0/24
```

`netmap-PID1.txt` can be cached at HTTP caches

3. Retrieve Cost Map (cacheable):

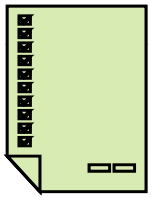
```
GET /alto/costmap-PID1.txt
```

```
r=getcost COST routing numerical
c=SRC PID1 DEST PID2 8
c=SRC PID1 DEST PID3 13
```

`costmap-PID1.txt` can be cached at HTTP caches

ALTO Server
(alto.isp.net:80)

ALTO
Info



P2P Client
(in PID1)

P2P Client

Peer
Exchange

Direct
Connect

DHT

Tracker

	Cost Mode (Ord./Num.)	Source Grouping	Destination Grouping	Transport/ Encoding	Caching of ALTO Info	P2P Redist.
H12		IP, IPPrefix	IP, IPPrefix	TCP/ encoding not specified yet		
Proxider/ Oracle	Ranking	Per [Src IP, list of Dst IP]; Mentioned possibility to replace IP by IPPrefix/ASN	IP address; Mentioned possibility to replace IP by IPPrefix/ASN	Not specified yet		
Client to Service Query Response Protocol for ALTO	Ordinal/ Numerical	Location context specifying precision level; Host location attribute can use PID for aggregation	Location context specifying precision levels; Host location attribute can use PID for aggregation	XML/MIME type (application/p2p-alto+xml); uri specifying service location; enroll to get config.; binary		
ALTO (P4P/ InfoExport)	Ordinal/ Numerical ; Numerical important for Primal/Dual Optimization Decomp. and completeness	Flexible grouping including IP , IP Prefix, ASN, PID	Flexible grouping including IP, IP Prefix, ASN, PID	HTTP/Text (application/ alto); interface descriptor to specify configuration	HTTP Caching	Yes

Backup Slides

Example 3: ALTO Client Embedded in a P2P Client using Ranking

ALTO Server
(alto.isp.net:80)

ALTO Client
@P2P Client
(in PID1)

1. Request Interface Descriptor

```
i=getcost                          /alto/msg.cgi?q=cost
i=getcost-source                   /alto/cost-PID1.txt
i=getcost-complete                 /alto/cost-complete.txt
i=getnetworkmap                    /alto/msg.cgi?q=netmap
i=getnetworkmap-source             /alto/netmap-PID1.txt
i=getnetworkmap-complete           /alto/netmap-complete.txt
p=PID1
```

2. Request Ranking:

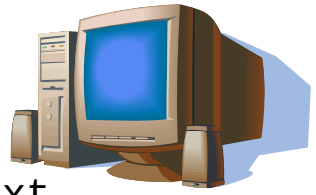
```
POST /alto/msg.cgi?q=cost HTTP/1.1
q=getcost COST routing ordinal
C=SRC IP0 DEST IP1 IP2 … IP100
```

```
r=getcost COST routing ordinal
c=SRC IP0
IP20 1
IP11 2
…
IP42 50
```

Examples with Animations

ALTO Server
(alto.isp.net:80)

ALTO Client
@appTracker



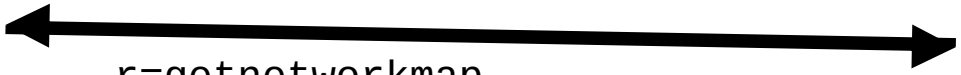
Request Interface Descriptor

```
i=getcost                  /alto/msg.cgi?q=cost
i=getcost-complete         /alto/cost-complete.txt
i=getnetworkmap            /alto/msg.cgi?q=netmap
i=getnetworkmap-complete   /alto/netmap-complete.txt
…
```

ALTO Server
(alto.isp.net:80)

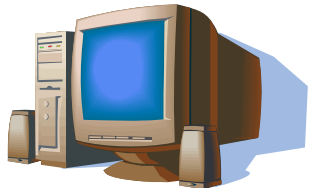
Retrieve Network Map (cacheable):

```
GET /alto/netmap-complete.txt
```



```
r=getnetworkmap
m=PID1 128.36.0.0/16
m=PID2 130.132.0.0/16
m=PID3 65.198.30.0/24
```

ALTO Client
@appTracker



`netmap-complete.txt` may be cached at HTTP caches

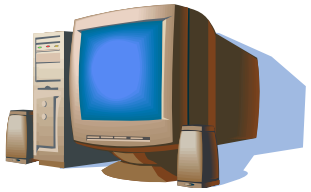
ALTO Server
(alto.isp.net:80)

Retrieve Cost Map (cacheable):

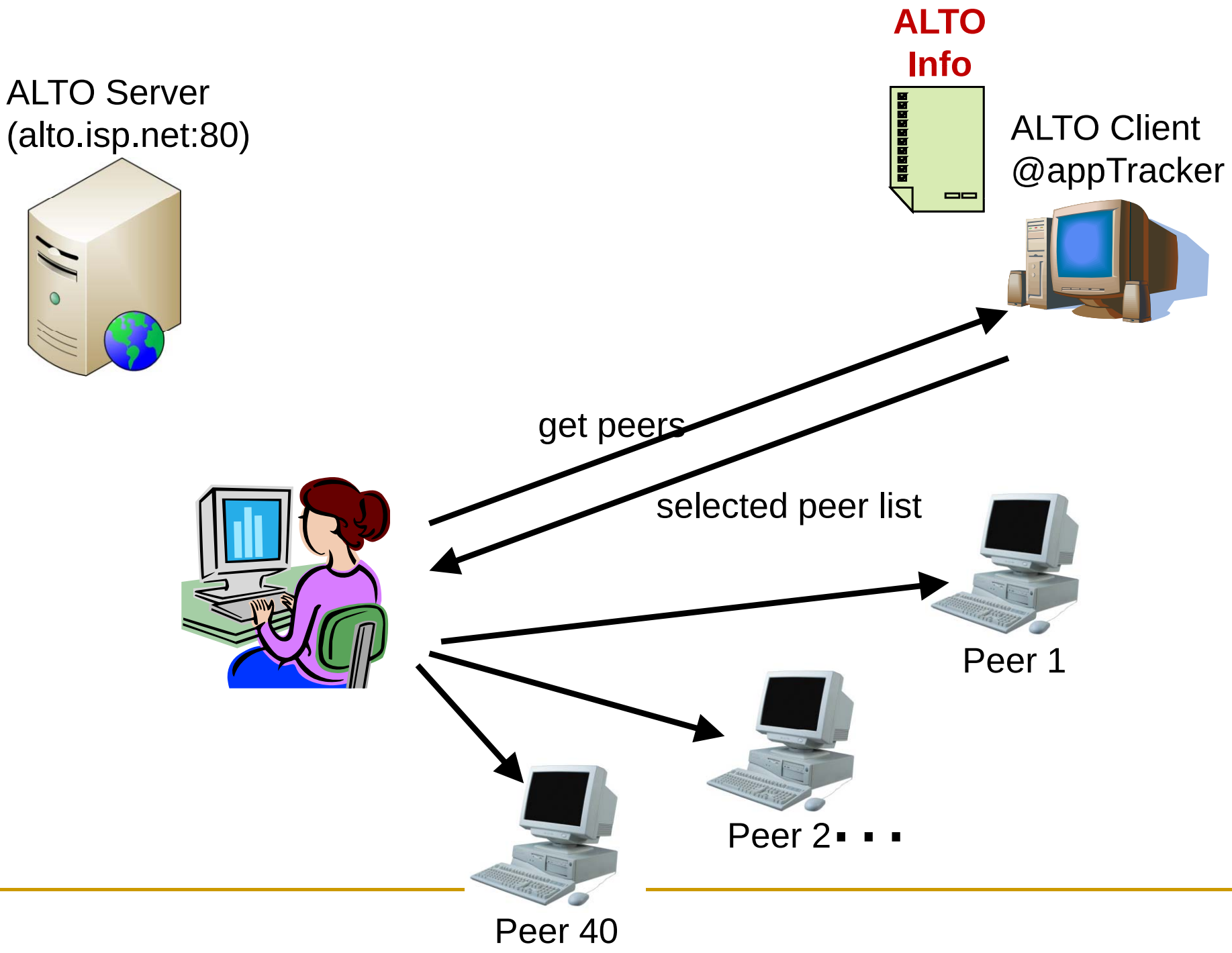
```
GET /alto/costmap-complete.txt
```

```
r=getcost COST routing numerical
c=SRC PID1
DEST PID1 1
DEST PID2 8
DEST PID3 3
c=SRC PID2
…
```

ALTO Client
@appTracker



`cost-complete.txt` may be cached at HTTP caches


ALTO Server
(alto.isp.net:80)
ALTO
Info
ALTO Client
@appTracker
get peers
selected peer list
Peer 1
Peer 2
Peer 40

Example 2: ALTO Client Embedded in a P2P Client

ALTO Server
(alto.isp.net:80)

P2P Client
(in PID1)

Request Interface Descriptor

```
i=getcost                   /alto/msg.cgi?q=cost
i=getcost-source            /alto/cost-PID1.txt
i=getcost-complete          /alto/cost-complete.txt
i=getnetworkmap             /alto/msg.cgi?q=netmap
i=getnetworkmap-source      /alto/netmap-PID1.txt
i=getnetworkmap-complete    /alto/netmap-complete.txt
p=PID1
```

ALTO Server
(alto.isp.net:80)

Retrieve Network Map (cacheable):

```
GET /alto/netmap-PID1.txt
```

```
r=getnetworkmap
m=PID1 128.36.0.0/16
m=PID2 130.132.0.0/16
m=PID3 65.198.30.0/24
```

P2P Client
(in PID1)

`netmap-PID1.txt` may be cached at HTTP caches

ALTO Server
(alto.isp.net:80)

Retrieve Cost Map (cacheable):

```
GET /alto/costmap-PID1.txt
```

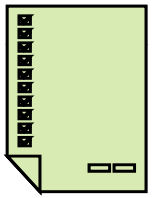
```
r=getcost COST routing numerical
c=SRC PID1 DEST PID2 8
c=SRC PID1 DEST PID3 13
```

P2P Client
(in PID1)

`costmap-PID1.txt` may be cached at HTTP caches

ALTO Server
(alto.isp.net:80)

ALTO Info



P2P Client
(in PID1)

Peer
Exchange

P2P Client

new
peers

new
peers

DHT

Tracker

Example 3: ALTO Client Embedded in a P2P Client using Ranking

ALTO Server
(alto.isp.net:80)

P2P Client
(in PID1)

Request Interface Descriptor

```
i=getcost                   /alto/msg.cgi?q=cost
i=getcost-source            /alto/cost-PID1.txt
i=getcost-complete          /alto/cost-complete.txt
i=getnetworkmap             /alto/msg.cgi?q=netmap
i=getnetworkmap-source      /alto/netmap-PID1.txt
i=getnetworkmap-complete    /alto/netmap-complete.txt
p=PID1
```

ALTO Server
(alto.isp.net:80)

Request Ranking:

```
POST /alto/msg.cgi?q=cost HTTP/1.1
q=getcost COST routing ordinal
C=SRC IP0 DEST IP1 IP2 … IP100
```

```
r=getcost COST routing ordinal
c=SRC IP0
IP20 1
IP11 2
…
IP42 50
```

P2P Client
(in PID1)

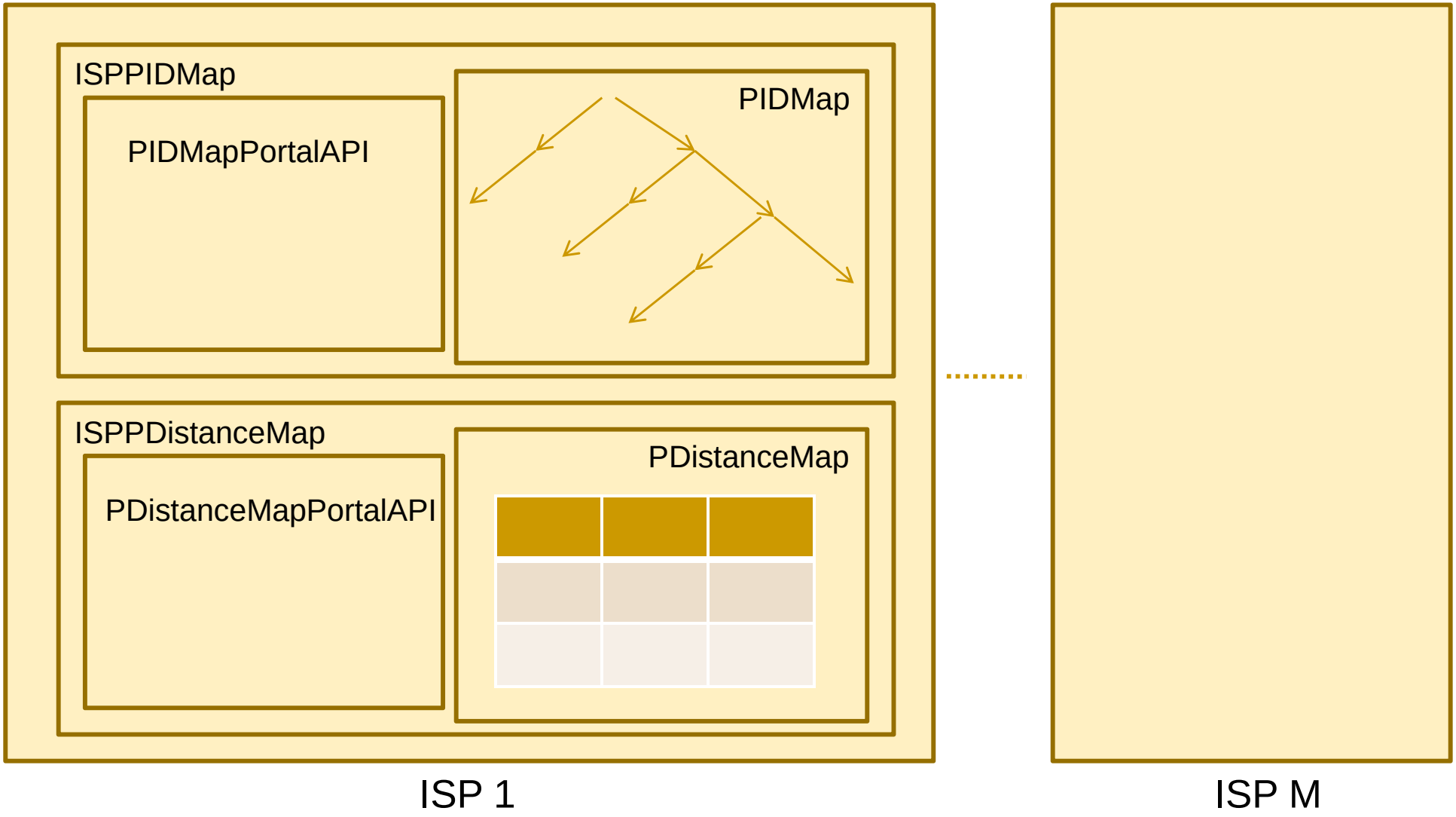
How May a P2P Application Use ALTO Information?

- This depends on the applications
- It is a place for application innovation

Example: Tracker-Based

- The tracker resolves the PIDs of clients
  - By using PID Maps
- The tracker uses a peering weight matrix to select initial peers for a new client
- Peering weight matrix computed according to channel state and pDistance matrix

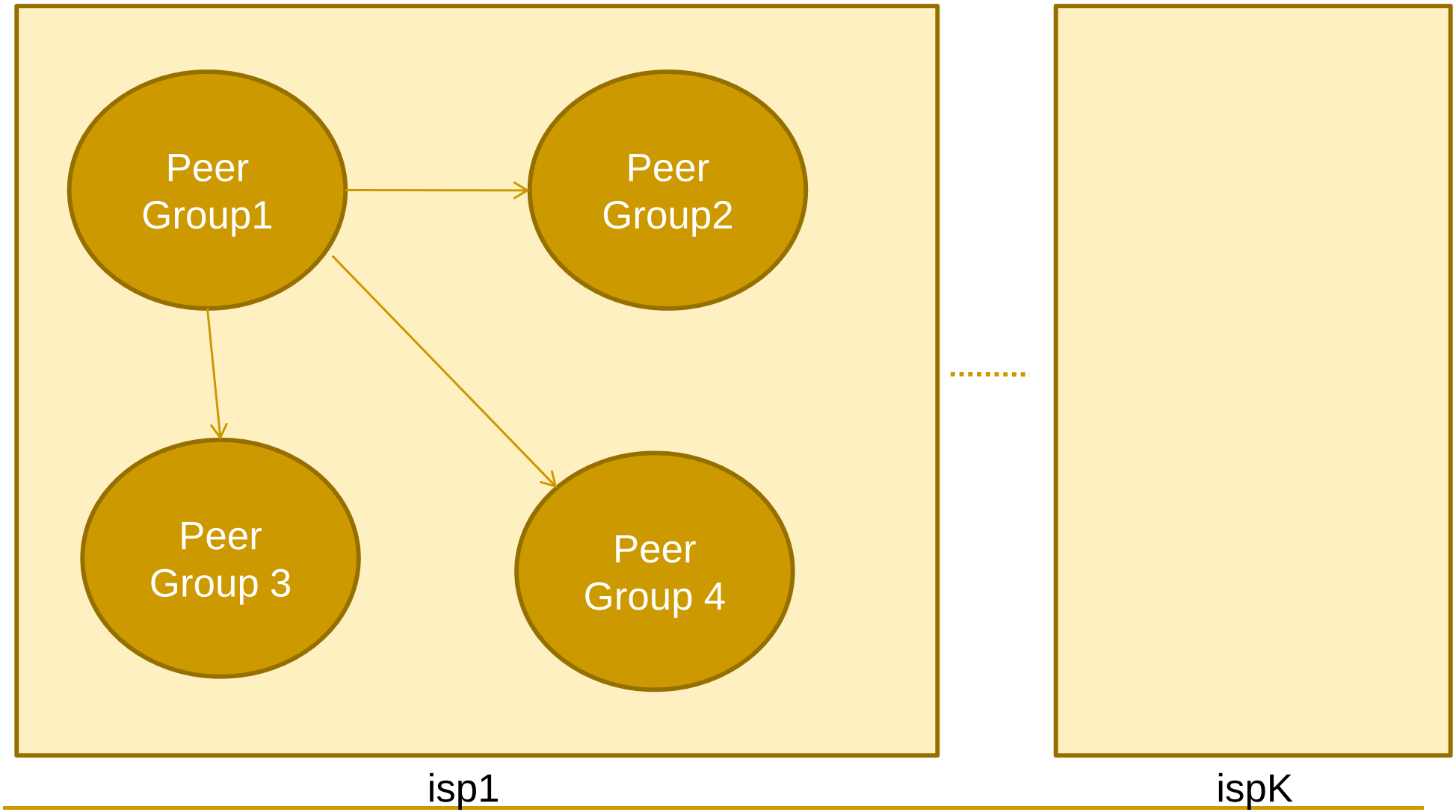
Tracker ISP Data Structures



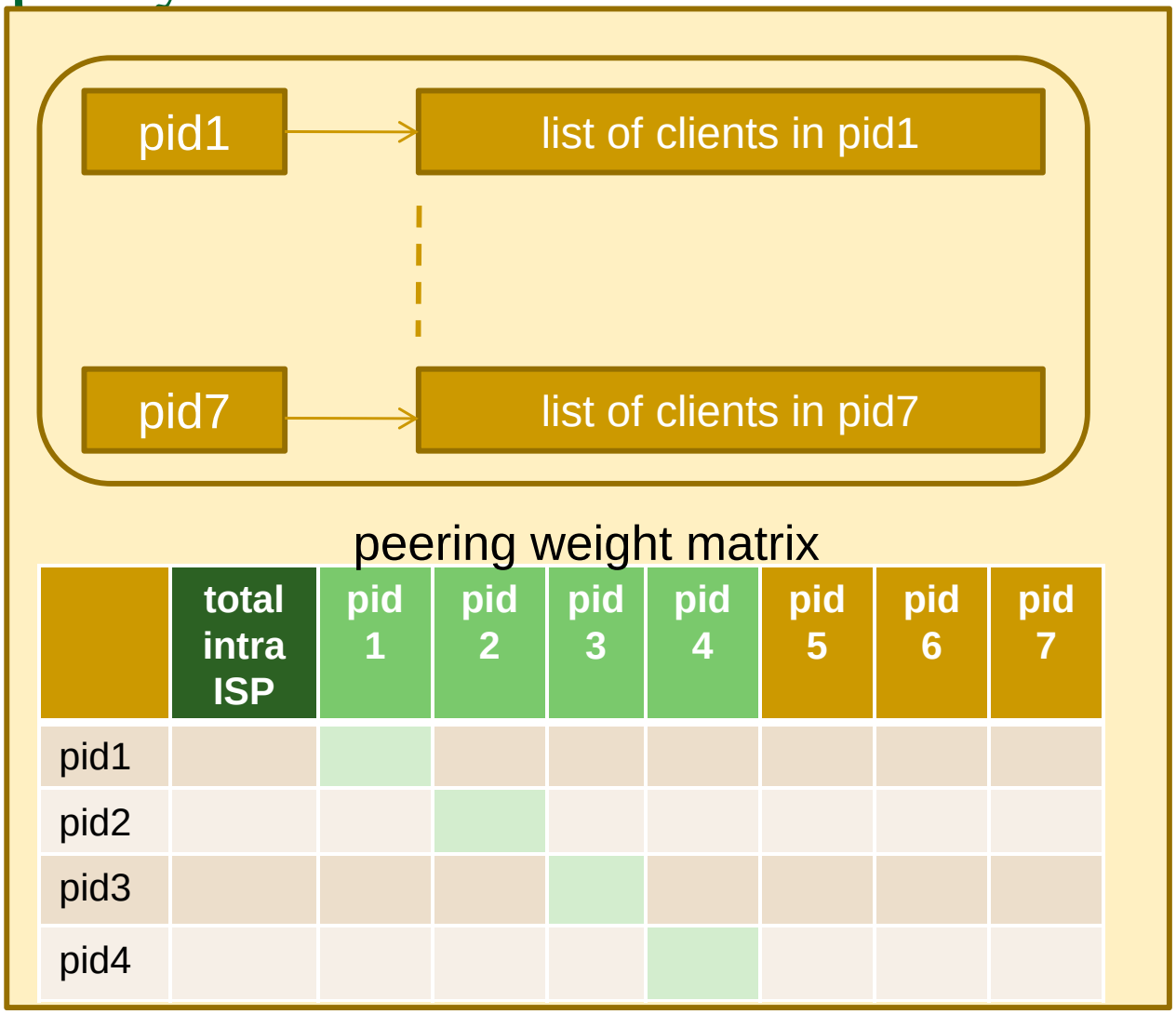
Per Channel Data Structures

- A list of ISPs maintained for the channel
- An ISPView for each maintained ISP
  - An ISP view partitions the peers in the channel according into peer groups
  - A PeeringGuidance matrix guides how to select peers from the peer groups,
    - e.g., for a peer in peer group 1, how many peers to select from group 1, from group 2, …

Per Channel Data Structures



Per Channel Data Structures: Peer Group by PID



peering weight matrix

	total intra ISP	pid 1	pid 2	pid 3	pid 4	pid 5	pid 6	pid 7
pid1								
pid2								
pid3								
pid4								