

ALTO  
Internet-Draft  
Intended status: Standards Track  
Expires: January 13, 2011

H. Song  
M. Chen  
Huawei  
Y. Wang  
Unknown  
D. Bryan  
Cogent Force, LLC / Huawei  
July 12, 2010

An Aggregate Network and Cost Map (CPID) Extension for the ALTO Protocol  
[draft-wang-alto-cpid-01](#)

Abstract

The goal of the Application-Layer Traffic Optimization (ALTO) protocol is to provide guidance to applications which have to select one or several hosts from a set of candidates, all of which are able to provide the desired resource. The goal of the mechanisms specified in this document is to extend the existing solution, and provide basic attributes to each end terminal to make it network-aware. These mechanisms introduce a way to directly transfer the guidance or costs using Network Location identifiers/PIDs, called CPIDs. And the existing The proposed solution inherits the existing solution's architecture, messages, and other mechanisms. It can be used as an independent solution or function together with other functions in the existing solution to provide guidance for different client and balance the workload on the server. Additional details for the process will be provided in the next version of the draft.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology and Concepts . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Overview . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	CPID Defined . . . . .	<a href="#">5</a>
<a href="#">3.1.1.</a>	What is a CPID? . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Protocol Overview . . . . .	<a href="#">6</a>
<a href="#">4.1.</a>	Messages . . . . .	<a href="#">6</a>
<a href="#">4.1.1.</a>	Capabilities Query . . . . .	<a href="#">7</a>
<a href="#">4.1.2.</a>	Guidance . . . . .	<a href="#">7</a>
<a href="#">5.</a>	Use Cases . . . . .	<a href="#">7</a>
<a href="#">5.1.</a>	ALTO Client Embedded in P2P Tracker . . . . .	<a href="#">8</a>
<a href="#">5.2.</a>	ALTO Client Embedded in P2P Client . . . . .	<a href="#">9</a>
<a href="#">6.</a>	Security Considerations . . . . .	<a href="#">10</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">10</a>
<a href="#">8.</a>	References . . . . .	<a href="#">10</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">10</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">10</a>
<a href="#">Appendix A.</a>	Examples . . . . .	<a href="#">11</a>
	Authors' Addresses . . . . .	<a href="#">16</a>



## **1. Introduction**

Many Internet applications, including widely used overlay applications for peer-to-peer file sharing and video streaming, need to access resources available in several equivalent replicas on different hosts, for example pieces of information or server processes. The goal of Application-Layer Traffic Optimization (ALTO) [[I-D.ietf-alto-problem-statement](#)] is to provide guidance to the applications, which have to select one or several hosts from a set of candidates able to provide the desired resource.

Participants have proposed a number of solutions to the ALTO WG, which have been merged to produce [[I-D.ietf-alto-protocol](#)]. The goal of the mechanisms specified in this document is to extend the existing solution to improve the performance of the protocol and better satisfy the privacy requirements. These mechanisms introduce a way to directly transfer the guidance or costs using Network Location identifiers/PIDs, called CPIDs. The proposed solution inherits the existing solution's architecture, messages, and other mechanisms. It can be used as an independent solution or function together with other functions in the existing solution to provide guidance for different client and balance the workload on the server. It can also reduce the risk regarding to the privacy issue, especially for P2P applications. This document is a work in progress and will be updated in the future.

## **2. Terminology and Concepts**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

This document also reuses the concepts defined in [[I-D.ietf-alto-problem-statement](#)] and [[I-D.ietf-alto-reqs](#)].

## **3. Overview**

The protocol proposed in this draft can be considered as an extension of or a complementary to the merged ALTO protocol, and inherits existing architecture from the draft [[I-D.ietf-alto-protocol](#)].



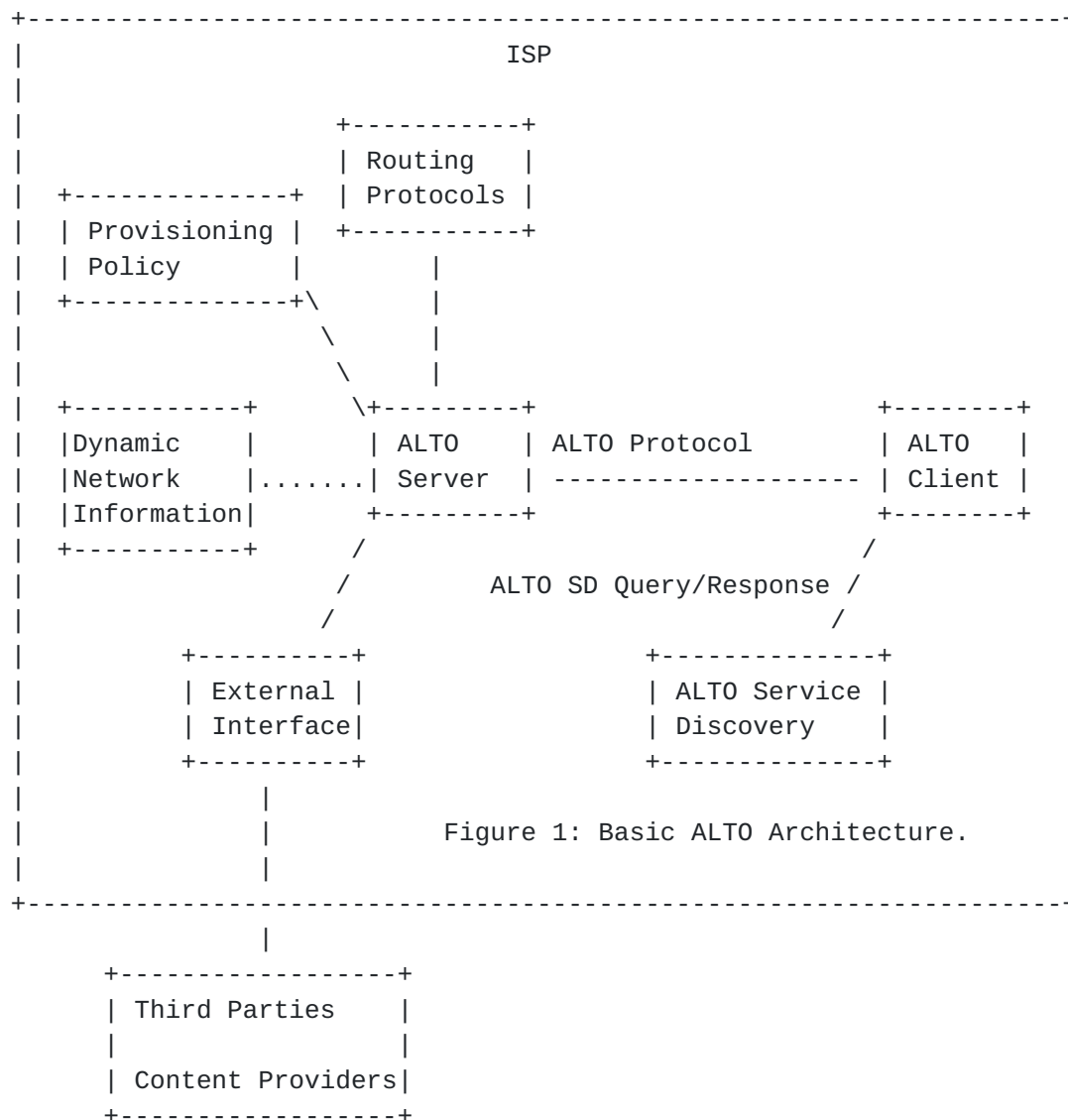


Figure 1: Basic ALTO Architecture.

## ALTO Architecture

Figure 1 (extracted from [[I-D.ietf-alto-protocol](#)]) shows the overall system architecture of the ALTO protocol. This draft is also consistent with this architecture. In this architecture, an ALTO Client uses ALTO Service Discovery to identify an appropriate ALTO Server; an ALTO Server prepares ALTO Information; and the ALTO Client requests available ALTO Information from the ALTO Server using the ALTO Protocol.

In our proposed ALTO protocol, the ALTO guidance messages are simplified to cost-peer-identifiers (CPID) request/response only. ALTO Information from the ALTO Server is abstracted and dispersed to each potential peer. When ALTO clients get the CPIDs, they also



obtain the ALTO guidance regarding those peers at the same time. As a result, requests and server workload, as well as privacy risks, are significantly reduced.

### **3.1. CPID Defined**

#### **3.1.1. What is a CPID?**

In [[I-D.ietf-alto-protocol](#)], the Network Map and the Cost Map are two separate data objects, even though the Cost Map is based on and reflects the relationship between the elements of the Network Map. If clients want to get ALTO guidance, they obtain both the the PIDs of candidate peers in the Network Map, as well as path ratings using the obtained PIDs. In our proposed extension, the Network Map and the Cost Map are combined into one map. The elements in this map act as new type of PID. As with the original PID in the network map, this new PID type, which we called CPID, provides an implicit way to specify a network aggregation. CPIDs reflect the costs/weights between peers indirectly. Assuming that CPID1 is the CPID of peer1, and CPID2 is the CPID of peer2, then cost between these two peers could be represented as the following equation.

$$\text{COST}(\text{peer1}, \text{peer2}) = \text{FUNC}(\text{CPID1}, \text{CPID2})$$

Here, the cost FUNC can be a simple multiplication, a cross-production or any formula that the ISP [QUESTION:should this be ISP, or ALTO server?] defines. After getting CPIDs of two peers, through this simple function ALTO clients can easily obtain the cost between these two peers for use as a path rating.

Both cost type and cost mode can use existing mechanisms. The cost Type attribute indicates what the cost represents. For example, an ALTO Server could define costs representing air-miles, hop-counts, or generic routing costs. The Mode attribute indicates how costs should be interpreted. For example, the cost can be interpreted as numerical values or ordinal rankings. Note that the type of CPID is independent of cost type and cost mode. The cost between two peers is calculated by the cost function with the two CPIDs as inputs, whatever the type of CPID is.

A CPID could be a simple binary code, a coordinates, or any other types defined freely. The type is decided by the method of CPID construction. The CPID construction method also decides the cost FUNC. The details of the construction method are not inside the scope of this draft. ISP may internally construct CPID using any method it chooses. A single CPID to a peer ostensibly is only an identifier no matter the type it chooses. When the CPIDs of the source peer and destination peers have been obtained, ALTO client





could calculate the costs for better peer selection.

#### **4. Protocol Overview**

The first step for a P2P application to use ALTO guidance is to discover one or more corresponding ALTO servers. ALTO Services and Servers are discovered through some certain mechanisms such as DNS, DHCP [[I-D.song-alto-server-discovery](#)]. After finding the correct ALTO server, ALTO client needs to know what the ALTO server could provide to assist its peer selection. By negotiation, ALTO Client will get a configuration file from a particular ALTO Server. The configuration file includes, for example, details about the type of CPID, cost FUNC, and cost metrics supported by the ALTO Server. Then, ALTO Client starts ALTO guidance requesting. In this protocol, the guidance for peer selection is derived through CPID of Endpoints. ALTO Client sends a Query to ALTO server for requesting CPID(s) of one or a set of Host Location Attributes (HLAs, [1]) of peers. ALTO Server considers the request and should reply with the corresponding CPID(s). When a P2P tracker acts as an ALTO client, it could retrieve the CPID for a peer when it joins and stores it locally for later use. If a P2P client requests for candidate source providers, P2P tracker will gather destination candidates and get the CPIDs of source and destination candidates locally. Then, P2P tracker calculates the costs of source and destination pairs by using cost FUNC with their CPIDs as the inputs. Finally, P2P tracker could determine the candidates to select according to the sorted costs. When a P2P client acts as an ALTO client, it could retrieve the CPID for itself and stores it locally for later use. P2P Applications could adopt CPID as an attribute of the application. The CPID could take part in the process of peer selection. When P2P client does resource discovery, it also gathers their CPIDs at the same time, e.g. the DHT will tell the requester the CPIDs of destination peers. Then P2P client could calculate and compare the costs, and determine the candidates to select finally. Note that the CPID should have a part to distinguish the ISP it belongs to. For the CPIDs in the same ISP, P2P client/trackers could directly calculate the cost of the path between the source peer and the destination peer. If the candidate peers in the same ISP are not enough, P2P client/tracker could use the ways in other drafts [[I-D.ietf-alto-protocol](#)] for guidance requesting, or directly use an ISP priority map for crossed ISP peer selection.

##### **4.1. Messages**



#### **4.1.1. Capabilities Query**

For seeking guidance in Resource Provider selection, an ALTO Client sends a Capability query to one or more ALTO Servers to discover which rating criteria are supported and at which accuracy level. Through some certain mechanisms such as DNS, DHCP [[I-D.song-alto-server-discovery](#)], ALTO Services and Servers could be discovered.

The capability query and response could be the same as other ALTO protocols. Through the queries and responses between servers and clients, ALTO server and client could negotiate about the PID type and cost FUNC for generating the cost from CPIDs, and may also about which rating criteria and which accuracy level for the ALTO server to adopt to provide CPID to ALTO client.

#### **4.1.2. Guidance**

In this solution, the guidance for peer selection is through CPID of Endpoints. In draft [6], the Endpoint Property Lookup query allows an ALTO Client to query for the properties of Endpoints known to the ALTO Server. CPID is also a type of PID, so we can reuse the Endpoint Property Lookup query to retrieve the CPID of each endpoint.

An ALTO Client sends an Endpoint Property Lookup query to an ALTO Server to request the guidance for peer selection. When a P2P tracker acts as an ALTO client, the query may include one or a set of HLAs of peers. This HLA represents new joint peer or the peer whose CPID need to be update. When a P2P client acts as an ALTO client, the query may include HLA itself or set null only.

The response from ALTO server should provide a CPID for each peer in the associated query. ALTO Client will store the CPID(s) locally for later use, and may also maintains a back-end database that updates at pre-determined intervals or sudden changes.

A Server may contain the used response cost FUNC in order to further assist the client to generate the cost for peer selection. It also could be retrieved in other ways. For example, the returned documents including cost FUNC can be downloaded by ALTO Clients or provisioned into devices.

## **5. Use Cases**



### 5.1. ALTO Client Embedded in P2P Tracker

A P2P tracker could handle a large number of clients. When a P2P tracker acts as an ALTO client to enable more network-efficient traffic patterns and improve application performance, the P2P tracker should request the ALTO information (CPIDs) for peers.

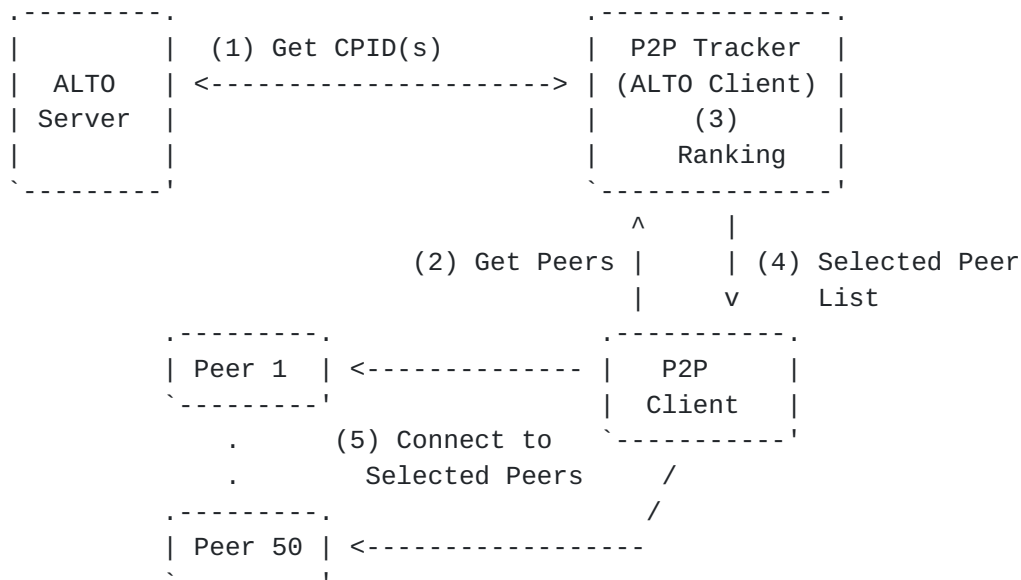


Figure 2 ALTO Client Embedded in P2P Tracker

Figure 2 shows an example that a P2P tracker as an ALTO Client uses ALTO information for peer selection. The example process is shown as follows:

1. When A P2P Client joins the swarm or need to be update, the P2P Tracker requests and obtains its CPID and then stores it locally. The P2P Tracker could collect new joint peers or the peers need to be updated together for CPID requesting.
2. A P2P Client requests a peer list from the P2P Tracker.
3. The P2P Tracker gathers destination candidates and gets the CPIDs of source and destination candidates locally. In case of sufficient candidate peers in the same ISP, the P2P Tracker then runs the COST-CALCULATE function to rank the candidate peers using their CPID. If not, P2P tracker could use the ways in other drafts [\[I-D.ietf-alto-protocol\]](#) for guidance requesting, or directly use an ISP priority map for crossed ISP peer selection.
4. The P2P Tracker returns the selected peer list to P2P client.
5. The P2P Client connects to the selected peers.



## 5.2. ALTO Client Embedded in P2P Client

P2P clients may also utilize ALTO information themselves for better peer selection. When a P2P Client works as an ALTO client, it typically queries only the ALTO Server that belongs to its own ISP for its own CPID. P2P Applications could adopt CPID as an attribute of the application. This CPID could take part in the process of peer selection, such as resource discovery. When P2P client discovers the candidate peers, it also gathers their CPIDs at the same time.

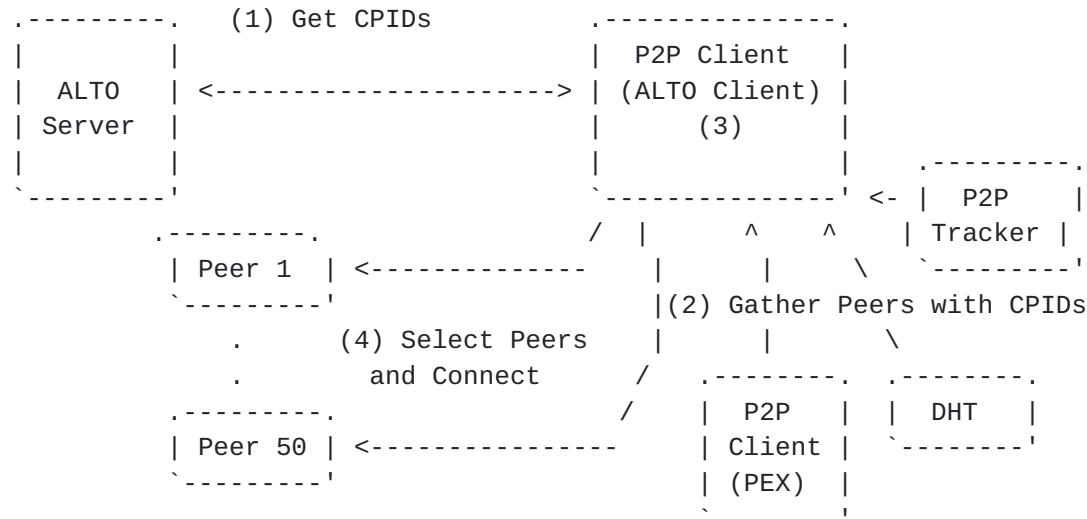


Figure 3: ALTO Client Embedded in P2P Client

Figure 3 shows an example use the case that a P2P client as an ALTO Client uses ALTO information for peer selection. The example process is shown as follows:

1. The P2P Client requests and gets its own CPID from its ALTO Server and then stores it locally
2. The P2P Client discovers peers together with their CPIDs from sources such as Peer Exchange (PEX) from other P2P Clients, Distributed Hash Tables (DHT), and P2P Trackers.
3. In case of sufficient candidate peers in the same ISP, the P2P client then runs the COST-CALCULATE function to rank the candidate peers using their CPID and determines the peers to be select according to the rank. If not, P2P client could use the ways in other drafts [[I-D.ietf-alto-protocol](#)] for guidance requesting, or directly use an ISP priority map for crossed ISP peer selection.
4. The P2P Client connects to the selected peers.





## **6. Security Considerations**

An ALTO Server may optionally use authentication and encryption to protect ALTO information for preventing the information being disclosed on the path. This will be further discussed in other documents.

## **7. IANA Considerations**

IANA may need to assign numbers to distinguish different ISPs with this solution.

## **8. References**

### **8.1. Normative References**

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

### **8.2. Informative References**

- [I-D.ietf-alto-problem-statement]  
Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", [draft-ietf-alto-problem-statement-04](#) (work in progress), September 2009.
- [I-D.ietf-alto-reqs]  
Kiesel, S., Previdi, S., Stiernerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", [draft-ietf-alto-reqs-05](#) (work in progress), June 2010.
- [I-D.akonjang-alto-proxidor]  
Akonjang, O., Feldmann, A., Previdi, S., Davie, B., and D. Saucez, "The PROXIDOR Service", [draft-akonjang-alto-proxidor-00](#) (work in progress), March 2009.
- [I-D.wang-alto-p4p-specification]  
Wang, Y., Alimi, R., Pasko, D., Popkin, L., and Y. Yang, "P4P Protocol Specification", [draft-wang-alto-p4p-specification-00](#) (work in progress), March 2009.
- [I-D.ietf-alto-protocol]



Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol",  
[draft-ietf-alto-protocol-04](#) (work in progress), May 2010.

[I-D.song-alto-server-discovery]

Yongchao, S., Tomsu, M., Garcia, G., Wang, Y., and V.  
 Avila, "ALTO Service Discovery",  
[draft-song-alto-server-discovery-03](#) (work in progress),  
 July 2010.

## Appendix A. Examples

To reflect the relationships of peer pairs exactly, we can use an n-dim vector to represent the CPID. The basic idea is that each node (can be a peer or an aggregated peer group) in an n-dimension hyperplane constructed by n nodes can be represented by an n-dimension coordinate at least. We already have all the distance/cost values between pair of nodes. Thus, we can formulate an n\*n dimension weight matrix W, where ELEMENT<sub>ij</sub> means the weight value from node i to node j. Presumably, ELEMENT<sub>ij</sub> in the weight matrix W can be considered as the inner product of node i and node j. Thus, we can find an n-dimension vector X to represent each node through matrix decomposition  $W = C \cdot C^T$ .

Assuming that we have n nodes needed to be assigned CPID and we have obtained the weights of all node pairs, we can formulate a weight matrix W as shown in Figure 4. The row suffix i and the column suffix j of each element W<sub>ij</sub> represent the source and destination of a path with the weight value or priority value W<sub>ij</sub>. The priority value can be an ordinary number from 1 to n or other weighted number according to the sorting of weight value for the same source node i. The diagonal elements which mean the source node and the destination node are the same have the highest weight or priority, if the highest weight or priority means the nearest or lowest cost.

	N1	N2	...	Nn	
/					\
N1	W11	W12	...	W1n	
N2	W21	W22	...	W2n	
.	...	...	...	...	
.	...	...	...	...	
Nn	Wn1	Wn2	...	Wnn	
\					/

Figure 4. Matrix of weight/priority



There are many ways to obtain a matrix  $C$  satisfying  $W = C \cdot C^T$ . The simplest way is Cholesky decomposition. Cholesky decomposition is a decomposition of a symmetric, positive-definite matrix into a lower triangular matrix and the transpose of the lower triangular matrix. The necessary condition for applying Cholesky decomposition is that the matrix  $W$  is symmetric and positive definite. Assuming that the weights of two directions between two nodes are the same and the weight in same node is high enough, these conditions can be satisfied. Then we can easily apply Cholesky decomposition on  $W$ , as shown in Equation. 1. The  $i$ -th row of matrix  $C$  is the coordinates of node  $i$  as the CPID.

$$\begin{array}{c}
 \begin{array}{c} / \\ | \quad W_{11} \quad W_{12} \quad \dots \quad W_{1n} | \\ | \\ W = | \quad W_{21} \quad W_{22} \quad \dots \quad W_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad W_{n1} \quad W_{n2} \quad \dots \quad W_{nn} | \\ \backslash \end{array} \\
 \\
 \begin{array}{c} / \\ | \quad C_{11} \quad C_{12} \quad \dots \quad C_{1n} | \\ | \\ = | \quad C_{21} \quad C_{22} \quad \dots \quad C_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad C_{n1} \quad C_{n2} \quad \dots \quad C_{nn} | \\ \backslash \end{array} * \begin{array}{c} / \\ | \quad C_{11} \quad C_{12} \quad \dots \quad C_{1n} | \\ | \\ | \quad C_{21} \quad C_{22} \quad \dots \quad C_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad C_{n1} \quad C_{n2} \quad \dots \quad C_{nn} | \\ \backslash \end{array}^T = C * C^T
 \end{array}$$

Equation. 1

When selecting peers in the same ISP for peer  $A$ , what we need to do is only calculating the inner products of CPIDs of peer  $A$  and other peers. And then sort the inner products and choose peers with high inner product values.

However, a problem of these above methods is the assumption that the weights from node  $i$  to  $j$  and from  $j$  to  $i$  should be same. Although it is the true in most cases, we still want to control or distinguish the weight  $W_{ij}$  and  $W_{ji}$ . The basic idea is to separate the coordinates of node to two parts. For node  $i$ , part of coordinate is the identifier  $C_{soui}$  as source, and the other is the identifier  $C_{desi}$  as destination. Therefore,  $W_{ij} = C_{soui} * C_{desj}^T$  is different from  $W_{ji} = C_{souj} * C_{desi}^T$ . Similarly, what we want here is not a coordinate matrix  $C$  but a source coordinate matrix  $C_{sou}$  and a destination matrix  $C_{des}$ . And the coordinate of node  $i$  is the combination of the  $i$ -th row of  $C_{sou}$  and the  $i$ -th row of  $C_{des}$ . There



are lots of matrix decomposition methods for finding  $C_{\text{sou}}$  and  $C_{\text{des}}$  satisfying  $W = C_{\text{sou}} * C_{\text{des}}^T$  such as LU, QR, Singular value decomposition (SVD). LU decomposition is a matrix decomposition which writes a matrix as the product of a lower triangular matrix and an upper triangular matrix. Cholesky decomposition is a special case of the symmetric LU decomposition, with  $L = U^T$ . Here we adopt SVD which is more efficiency and easier to extend. SVD is an important factorization of a rectangular real or complex matrix, with several applications in signal processing and statistics. Through applying SVD on  $W$ , we could obtain an equation  $W = U * S * V^T$ , where  $U$  is an  $n$ -by- $n$  unitary matrix,  $S$  is an  $n$ -by- $n$  diagonal matrix with nonnegative real numbers on the diagonal, and  $V^T$ , an  $n$ -by- $n$  unitary matrix, denotes the conjugate transpose of  $V$ . Usually the diagonal entries  $S_{i,i}$  are ordered in non-increasing fashion. In this case, the diagonal matrix  $S$  is uniquely determined by  $W$  (though the matrices  $U$  and  $V$  are not). The diagonal entries of  $S$  are known as the singular values of  $W$ . Through matrix transformation (choose freely) we could get the  $C_{\text{sou}}$  and  $C_{\text{des}}$  as shown in Equation. 2.





$$\begin{aligned}
 & \begin{array}{c} / \\ | \quad W_{11} \quad W_{12} \quad \dots \quad W_{1n} | \\ | \\ W = | \quad W_{21} \quad W_{22} \quad \dots \quad W_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad W_{n1} \quad W_{n2} \quad \dots \quad W_{nn} | \\ \backslash \end{array} \\
 \\
 & \begin{array}{c} / \qquad \qquad \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad T \\ | \quad U_{11} \quad U_{12} \quad \dots \quad U_{1n} | \quad | \quad S_{11} \quad 0 \quad \dots \quad 0 | \quad | \quad V_{11} \quad V_{12} \quad \dots \quad V_{1n} | \\ | \\ = | \quad U_{21} \quad U_{22} \quad \dots \quad U_{2n} | * | \quad 0 \quad S_{22} \quad \dots \quad 0 | * | \quad V_{21} \quad V_{22} \quad \dots \quad V_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad U_{n1} \quad U_{n2} \quad \dots \quad U_{nn} | \quad | \quad 0 \quad 0 \quad \dots \quad S_{nn} | \quad | \quad V_{n1} \quad V_{n2} \quad \dots \quad V_{nn} | \\ \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad / \end{array} \\
 \\
 & \begin{array}{c} / \qquad \qquad \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad T \\ | \quad U_{11} \quad U_{12} \quad \dots \quad U_{1n} | \quad | \quad V_{11} * S_{11} \quad V_{12} * S_{22} \quad \dots \quad V_{1n} * S_{nn} | \\ | \\ = | \quad U_{21} \quad U_{22} \quad \dots \quad U_{2n} | * | \quad V_{21} * S_{11} \quad V_{22} * S_{22} \quad \dots \quad V_{2n} * S_{nn} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad U_{n1} \quad U_{n2} \quad \dots \quad U_{nn} | \quad | \quad V_{n1} * S_{11} \quad V_{n2} * S_{22} \quad \dots \quad V_{nn} * S_{nn} | \\ \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad / \end{array} \\
 \\
 & \begin{array}{c} / \qquad \qquad \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad T \\ | \quad Csou_{11} \quad Csou_{12} \quad \dots \quad Csou_{1n} | \quad | \quad Cdes_{11} \quad Cdes_{12} \quad \dots \quad Cdes_{1n} | \\ | \\ = | \quad Csou_{21} \quad Csou_{22} \quad \dots \quad Csou_{2n} | * | \quad Cdes_{21} \quad Cdes_{22} \quad \dots \quad Cdes_{2n} | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad \dots \quad \dots \quad \dots \quad \dots | \quad | \quad \dots \quad \dots \quad \dots \quad \dots | \\ | \quad Csou_{n1} \quad Csou_{n2} \quad \dots \quad Csou_{nn} | \quad | \quad Cdes_{n1} \quad Cdes_{n2} \quad \dots \quad Cdes_{nn} | \\ \backslash \qquad \qquad / \qquad \qquad \backslash \qquad \qquad / \end{array} \\
 \\
 & \begin{array}{c} T \\ = \quad Csou \quad * \quad Cdes \end{array}
 \end{aligned}$$

Equation. 2

This multi-dimension CPID can be also performed in a hierarchical construction manner. We obtain the Csou and Cdes matrixes from weight or priority matrix of each level, and combine the ID of each level as the final representation. Noticed that if we want calculate the inner product of combined IDs directly, the weights or priorities in the high level should be larger enough than the low level to distinct different levels.



When peer A wants to select peers, it can send request only using the source part of CPID. The P2P tracker calculates the inner products of the source part of peer A's CPID with the destination part of CPIDs of other candidate peers in the same ISP. And then sort the inner products and choose the peers with high inner product values. It is an easy way to keep the secrecy.

Representing CPID with large dimensional data may be inconvenient and lead to low efficiency in message transmission and priority calculation, since the number of nodes needed to be assigned to an CPID is too big. At the same time, there is much redundant information in the weight matrix or n-dimension coordinates, since the network design usually follows some strategies and rules to some extent. Moreover, the consistency between different groups of nodes needs to be guaranteed. Therefore, a more efficient peer representation can be found. A common way is to reduce the dimensionality to an appropriate number. There are several methods for dimension reduction, including principal components analysis (PCA), projection pursuit, principal curves, self-organizing maps, as well as provides neural network implementations of some of the reviewed statistical models. We can use PCA for the dimension reduction because PCA allows each node's coordinate values to be represented by a set of uncorrelated low dimensional parameters and can minimize reconstruction error optimally under the L2 norm. In our case, we can apply the PCA to process the weight matrix decomposition directly, which can minimize the reconstruction error of weight matrix than applying PCA to coordinate matrix. By choosing the first  $r$  ( $r \leq n$ ) principal components (PCs) from PCA transform, we can reduce the size of the matrix significantly without losing much information of the topology.

$$W^{n \times n} = U^{n \times n} * S^{n \times n} * V^{n \times n^T} = U^{n \times r} * S^{r \times r} * V^{r \times n^T}$$

Equation. 3

$$C_{new-sou} = \begin{pmatrix} / & & & \backslash \\ | & U_{11} & U_{12} & \dots & U_{1r} | \\ | & & & & | \\ | & U_{21} & U_{22} & \dots & U_{2r} | \\ | & \dots & \dots & \dots & \dots | \\ | & \dots & \dots & \dots & \dots | \\ | & U_{n1} & U_{n2} & \dots & U_{nr} | \\ \backslash & & & & / \end{pmatrix}$$

Equation. 4



Then the ratio  $E_r$  (Equation. 7) indicates how much information is retained by projecting the vectors on the optimal  $r$ -dimensional hyperplane. In our test, the average of  $E_r = 90\%$  of weight relationships of network can be achieved only using 8 PCs, by which the reconstructed weight shows very small difference from the original weight.



Authors' Addresses

Haibin Song  
Huawei

Email: haibinsong.cn@gmail.com

Mach(Guoyi) Chen  
Huawei  
KuiKe Building, No.9 Xinxu Rd. Hai-Dian District  
Beijing 100085  
China

Email: mach@huawei.com

Yan Wang  
Unknown

Email: haibinsong.cn@gmail.com

David A. Bryan  
Cogent Force, LLC / Huawei

Email: bryan@ethernet.org



