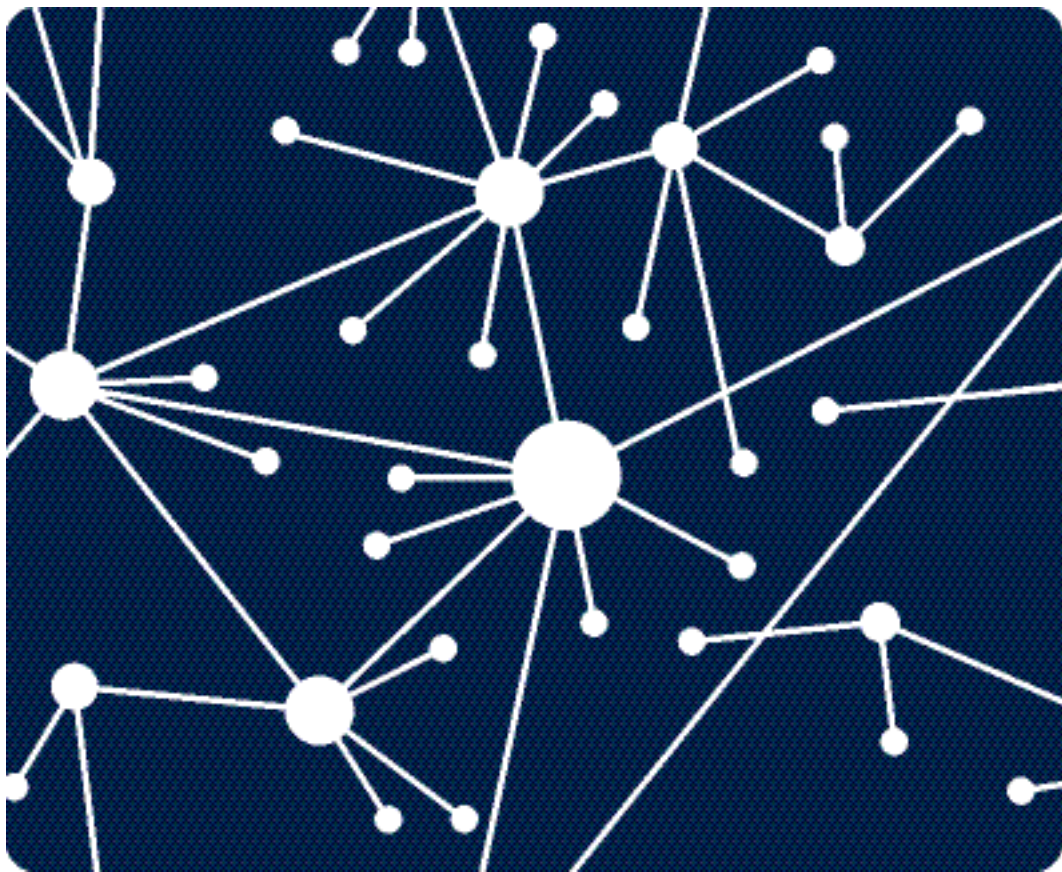




# Resilient Mobile IP

## *CoCo Architecture White Paper*



---

**Confidential and Proprietary** This document includes data that shall not be duplicated, used, or disclosed—in whole or in part—for any purpose other than for evaluation. The data subject to this restriction are contained in sheets 1-18.

---



*White Paper*  
*Resilient Mobile IP*  
*CoCo Architecture White Paper*

15 February 2008

**CoCo Communications Corporation**

[www.cococorp.com](http://www.cococorp.com)

999 3<sup>rd</sup> Avenue, Suite 3700  
Seattle, WA 98104

Phone: 206-284-9387

Fax: 206-770-6461

Copyright © 2002-2008 CoCo Communications Corporation.

CoCo is a trademark of CoCo Communications Corporation.

All Rights Reserved. Patents Pending.

The names of actual companies or products mentioned herein may be the trademarks of their respective owners.



# Table of Contents

|   |    |
|---|----|
| Introduction .....                              | 1  |
| About this document .....                       | 1  |
| Motivation .....                                | 1  |
| Quality and Type of Service .....               | 2  |
| Interoperability .....                          | 2  |
| Mobility and Rapid, Dynamic Configurations..... | 2  |
| Identity Security .....                         | 3  |
| Network Security .....                          | 3  |
| Scalability .....                               | 3  |
| Addressing in CoCo Networks .....               | 4  |
| Architectural Overview .....                    | 4  |
| Routing Layer.....                              | 5  |
| Recent Developments .....                       | 5  |
| Virtual Infrastructure.....                     | 5  |
| Clustering System Overview .....                | 5  |
| Clustering Terminology and Concepts .....       | 6  |
| A Clustering Example.....                       | 7  |
| Tree Representation of Clusters.....            | 8  |
| Clustered Route Advertisement.....              | 8  |
| Location-based Routing .....                    | 10 |
| Circuit Layer.....                              | 12 |
| Circuit Establishment .....                     | 12 |
| Circuit Tables .....                            | 13 |
| Circuit Layer Multipath Support .....           | 13 |
| Circuit Layer Multicast Support .....           | 15 |
| Identity Management.....                        | 16 |



## Introduction

The CoCo Protocol is a unique combination of enhancements to existing IP system design intended to increase usability, reliability, mobility, and security. It installs as a virtual network adapter on the whole range of Windows and Linux operating systems. Taken separately these techniques mirror state-of-the-art developments in mobile ad-hoc networking (MANET), fast IP mobility, peer-to-peer security, and media-independent handover (MIH). Together, they form an offering not otherwise available today.

That said, as the IETF working groups come to resolution on best practices for each of these components, CoCo is firmly committed to supporting open standard architecture. One current example is 802.21, which promises a sufficiently flexible and powerful set of primitives to be considered a solid improvement upon CoCo's private design. While this standard is still far from adoption, we are moving to align our internal structures for rapid compliance when that day comes. We believe that our network architecture holds its merit before, during, and after standardization of the key elements.

## About this document

This document is intended for a technical audience—including CoCo developers, CoCo technical sales staff, and the technical staff of CoCo business partners and customers. This document gives an overview of the functionality and design of the CoCo Protocol. It provides a high-level description of the protocol layers, the interfaces between them, and the interface the CoCo Protocol offers to the user level.

The “Motivation” section explains the motivation for the CoCo Protocol by highlighting the network features it supports that are unavailable with existing technology. The section “CoCo Protocol Layers” gives a summary description of the protocol layers. The section “Layer Features and Interfaces” explains each layer in more detail.

## Motivation

Data and voice communication systems play increasingly important roles in the military, government, and civilian sectors. Since the original development of the Transmission Control Protocol (TCP) and the Internet Protocol (IP) in the early 1970s, computer networks have advanced significantly. The simultaneous deregulation of conventional telephone systems spurred a wide array of new telephony services. Mobile cellular networks have also become increasingly sophisticated and widespread. Despite these developments, modern networks cannot provide adequate infrastructure for many critical applications such as first responder communication. The CoCo Protocol addresses shortcomings of existing network technologies; it enables modern and emerging communication technologies in the most demanding applications.

In the 1970s, robustness and fault-tolerance were key design goals for the TCP/IP protocols. The DARPA funding agencies were especially interested in networks that could maintain function by routing packets around nodes hit in a military strike and manage the resulting congestion.

Today's networks have additional requirements, including:

- **Quality of service:** the ability to support a wide variety of applications, including voice, video, and data
- **Interoperability:** the ability to use all common physical transport technologies and hardware devices
- **Dynamic, scalable routing:** the ability to support rapid user movement and ad-hoc network formation
- **Security:** the ability to authenticate users and resist network attacks

Some technologies attempt to implement some of these features by extending traditional transport protocols. CoCo's system architecture addresses all of these requirements.

## Quality and Type of Service

The ability to specify the quality of service (QoS) and type of service (ToS) is a recent development in network protocol design. QoS and ToS allow user applications to access an application program interface (API) that permits them to specify the intended use of a network path. For example, user programs may request a voice or data path. The Internet protocols, by comparison, were designed to support only data and so VOIP is built on the voice-over-data model. CoCo's implementation focuses on creating and maintaining voice circuits and uses those circuits to carry data traffic to reproduce the data-over-voice model so successfully deployed by 3GPP.

## Interoperability

The CoCo Protocol supports a wide variety of physical transport technologies including cellular, WiFi, Ethernet, and satellite—which enables construction of internetworks based on different underlying physical transports. For this reason, the CoCo Protocol is called an overlay protocol. CoCo technology fosters interoperability also because it is a pure software technology that runs on widely available, off-the-shelf hardware devices commonly used for wireless communication such as PDAs, cell phones, laptops, and wireless access points. A CoCo device, or CoCo node, is any computational device provisioned with the CoCo Protocol software. The section “The Physical Layer” discusses interoperability in more detail.

## Mobility and Rapid, Dynamic Configurations

All network devices may serve as routers in CoCo networks. As devices turn on and off, or move from one location to another, the network dynamically reconfigures without the intervention of network administrators. In

conventional cellular networks, towers are situated at fixed locations, so the network's routing resources are static and not configurable. CoCo networks work more flexibly and with a finer granularity of networking resources. In conventional cellular networks, each connected device draws upon a fixed supply of bandwidth. Since all CoCo devices can serve as routers when necessary, each device adds bandwidth resources and routing capability. Moreover, inexpensive wireless routers may serve as CoCo nodes and be easily repositioned to locations where more bandwidth is required; for example, an emergency site.

## Identity Security

The CoCo Protocol uses FIPS 140-2 cryptographic primitives to support identity validation and service authorization. The TCP/IP protocols do not address privacy and authentication, but leave these features for application developers to implement at the user level, so there is no uniform standard for Internet security. Conversely, security primitives are built directly into the CoCo Protocol on multiple levels to ensure consistency. Refer to the section "The Circuit Layer" for more detail.

Identity security has far-reaching consequences. For example, e-mail spam in its present form would be impossible since senders would be unable to forge their identities. Web servers on the Internet know only the external IP address of the source of each page request. The Internet protocols make no guarantees about the identity of a user. By contrast, a CoCo network server knows the identity of each user requesting a page. Protocol-level identity security allows advanced authorization technologies across the entire network.

## Network Security

The CoCo Protocol resists denial-of-service attacks, man-in-the-middle attacks, and traffic analysis attacks using best practice defenses including but not limited to secure pair-wise link keying and end-to-end bulk encryption. The topic of peer-to-peer security is beyond the scope of this document.

## Scalability

The CoCo Protocol scales effectively to large network sizes while maintaining connectivity and the ability to route packets efficiently in a dynamically changing network. This is a result of CoCo's novel addressing scheme and clustering mechanism. To avoid the need for all nodes to exchange messages with each other, which results in  $O(N^2)$  communication complexity in networks of size  $N$ , the CoCo Protocol decomposes the network into a hierarchy of regions called clusters. The section "The Routing Layer" explains addressing and clustering in more detail.

## Addressing in CoCo Networks

A key concept in the CoCo Protocol is the logical separation of a device's identity from its location. This advancement is echoed by modern developments in IEEE and IETF working groups, and is fundamental to creating a positive mobile IP experience. These groups recommend a move to IPv6 to achieve these goals. The CoCo implementation offers this functionality in existing IPv4 configurations.

Conceptually, CoCo provides media-independent handover between various physical transports without disturbing the IP stack. The implementation is transparent to existing IP networks, currently using UDP encapsulated source routing and in the future using IPv6 address translation. In the absence of a mobility provider service, legacy IPv4 machines can still be accessed transparently but the handover functionality reverts to traditional IP mode.

Traditional IP addresses refer to a specific interface rather than the host, thus each interface has a distinct address and can be thought of as a separate network location. This means that switching traffic from one interface to another effectively changes the identity of the connection and requires a complete session reconnect. However, IP allows for enough abstraction that we can use local private addresses to refer temporarily to remote hosts rather than one specific interface on that remote host. Network address translation allows for interoperation with the unmodified sender and receiver IP stacks.

The section "The Routing Layer" gives more information about CoCo locations.

## Architectural Overview

Externally, the CoCo stack fits ideally between existing OSI layer 2 and layer 3 implementations, where ARP currently resides. Internally, it divides into four layers: Routing, Circuit, Identity, and Addressing.

|                     |                   |                 |                  |
|---------------------|-------------------|-----------------|------------------|
| Address Translation |                   |                 |                  |
| Identity Management |                   |                 |                  |
| Circuit Routing     |                   |                 |                  |
| Packet Routing      |                   |                 |                  |
| Cluster<br>MANET    | Satellite<br>Data | Carrier<br>Data | Wi-Fi<br>Hotspot |

Figure 1: Conceptual Layers

## Routing Layer

The routing layer consists of several concrete transport objects with identical abstract interfaces. These transports generally divide into IP infrastructure and IP MANET ad-hoc types. They communicate with the various network media through standard host operating system network interface drivers, so most modern technologies are supported transparently. These transports provide abstract interfaces to simple management functions such as channel reservation, peer discovery, multicast group functions, and quality metrics such as round-trip time.

## Recent Developments

Historically, our products preferred the MANET point of view and so used encapsulation to extend the mesh over various infrastructure transports. Today's thinking reflects a substantial shift in that design, instead preferring the IP perspective by assigning temporary addresses to mesh peers.

This means that the implementation for Internet-connected infrastructure transports is simply a pass-through that allows IP routing to do what it does best.

## Virtual Infrastructure

When all infrastructure networks fail, devices powered by CoCo may fall back into virtual infrastructure mode. This unique offering enables common IP services such as DHCP and DNS for dynamic MANET configurations that scale up to thousands of devices without trouble. This technology is a hybrid of traditional mesh routing protocols with landmark-based communication reduction. Clustering decomposes a network into a hierarchy of regions in a manner analogous to the way cities and states provide a geographical hierarchy that facilitates addressing. The clustering mechanism assigns each network node a *location* based on the layers of clusters that contain it.

For the routing mechanism to scale efficiently, advertisements and locations cannot propagate completely through the network. (If they did, the number of messages exchanged in a network of  $N$  nodes would be  $O(N^2)$ .) A given node's location is not commonly known, and its advertisements are not sent to all other nodes. The clustering model controls the extent to which locations and advertisements propagate to limit message passing overhead. The routing system uses a mechanism called *location-based routing* which uses the best destination approximation contained in a node's location. The remainder of this section explains the details of these techniques.

## Clustering System Overview

The main purpose of the clustering system is to create a location-based addressing system that assists dynamic routing. One of the design goals of this system is to limit the number of messages required by the routing system as the network grows in size. It does this by limiting the propagation

distance of advertisements. The clustering system enables a node *S* to route data to a destination node *D* effectively, *even if the destination node D does not appear in the routing table of S*. This is the key benefit of *location-based routing*.

This section gives a high level, intuitive explanation of the clustering concept. A more mathematically rigorous development of clustering appears in the section “Clustering Graph Theory”.

The clustering mechanism defines *clusters* as regions in a network. It also assigns to each network device a *location* defined in terms of these clusters. The clustering mechanism defines a hierarchy of clustering levels. Before describing this in detail, it is helpful to consider a geographical analogy.

Regions such as country, state, county, and city determine geographic location as a series of increasingly precise refinements. Each of these regions contains distinguished cities that *represent* them, for example, capitols can represent states and county seats can represent counties. We may represent the geographic location of a city as a sequence of cities, each one the representative of a successively smaller region. For example, the location of the city Bellevue, Washington could be denoted by the ordered sequence of cities: [Washington D.C., Olympia, Seattle, Bellevue] since each represents one of the regions containing Bellevue:

- Washington D. C. “represents” America (a level 3 region)
- Olympia “represents” Washington state (a level 2 region)
- Seattle “represents” King county (a level 1 region)
- Bellevue “represents” itself (a level 0 region)

In this example, a node in London, England could send data to Bellevue, Washington without knowing the best route to Bellevue itself, simply knowing a good way to reach Washington D. C.

### **Clustering Terminology and Concepts**

The goal of the cluster hierarchy is to provide a way of specifying a network location for each node in a CoCo network. This section develops the necessary terminology and concepts.

A CoCo network *cluster* is a set of at least two nodes where at least one of the nodes is directly connected to each of the others. The clustering system takes an initially undifferentiated collection of nodes and assigns each to a distinct cluster. In each cluster, it also designates one of the nodes that is directly connected to all the others as the *cluster representative*.

Once the initial set of clusters is formed, the clustering mechanism may be applied to the clusters themselves. In this case, the original clusters are considered individual nodes, where each cluster representative stands for the cluster it represents.

Successive applications of this clustering process results in a hierarchy of cluster levels. See Definition 2 and Fact 3 in the section “Clustering Graph Theory” for a mathematically precise formulation of this process. Each cluster has at least two members, so the number of clusters at each level is at most half the number of nodes. Therefore, there are at most  $\log_2 N$  levels in a network with  $N$  nodes.

In the geographic analogy, the first level of clustering corresponds to the formation of counties from collections of towns, and the second level of clustering corresponds to the formation of states from collections of counties.

The *network location* of a device  $D$  is a sequence of cluster representatives  $[D_n, D_{n-1}, \dots, D_1, D_0]$ , where  $D = D_0$  and  $D_i$  is the representative of the cluster of level  $i$  which contains  $D$ . The smaller the value of the subscript  $i$ , the closer the distance from  $D_i$  to  $D_0$ . So the location of a CoCo node  $D$  may be viewed as a sequence of positions that converge to  $D$ , just as the set of cities in the geographic example is a sequence that converges on Bellevue.

### A Clustering Example

Figure 2: Clustering Example illustrates the clustering hierarchy in a simple network. Circles and ellipses indicate clusters; boldface borders indicate cluster representatives. There are two level 1 clusters, one containing X, Y, and Z, and another containing U, V, and W.

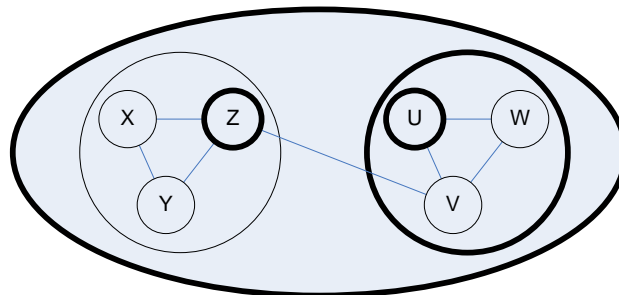


Figure 2: Clustering Example

The result of replacing clusters with their representatives in the clustering decomposition is the reduced network shown in Figure 3.

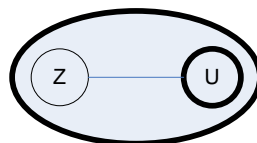


Figure 3: Clustering Example, continued: Level 1 Clusters

Repeating this process again yields a single node, as shown in Figure 4:



Figure 4: Clustering Example, continued: Level 2 Cluster

This example illustrates the concepts of the formation of clusters and a of cluster-level hierarchy.

### Tree Representation of Clusters

The tree in Figure 5 represents the cluster hierarchy for the network of Figure 2. In this tree, each non-leaf node represents a cluster, and the children of that node represent the cluster's members. For example at level 1 node Z represents a cluster containing X, Y, and Z.

Conceptually, a node X's location contains a sequence of network locations that become progressively closer to X. The section "Location-based Routing" explains how the location concept contributes to routing scalability.

Let the *height* of a node X be the distance of the shortest path from X to a leaf node (so that leaf nodes have height 0, parents of leaf nodes have height 1, and so on). The *rank* of a node in a network is defined as the height of the highest node it occurs in the cluster tree. Equivalently, the rank of a node is the level of the highest cluster it represents.

In these figures, the nodes X, Y, V, and W have rank 0, the node Z has rank 1, and node U has rank 2. The network location of a node may be obtained by following the sequence of nodes along the path from the root of the cluster tree to the leaf that represents that node.

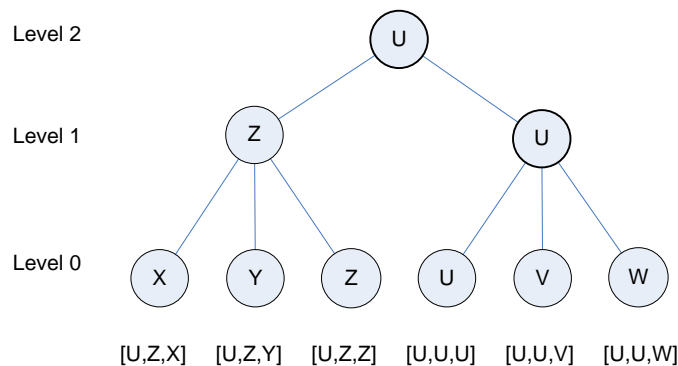


Figure 5: Cluster Tree with Network Locations

For example, in Figure 5, the location of X is [U, Z, X], since

- X is its own level-0 cluster
- X is part of a level-1 cluster whose representative is Z
- Z is part of a level-2 cluster whose representative is U

### Clustering Graph Theory

This section gives the mathematical background that is the basis for the clustering concepts. The goal of this section is to understand what clusters are and to understand how the clustering levels give rise to network locations.

Let  $G = \langle V, E \rangle$  be a graph where  $V$  is the set of vertices of  $G$  and  $E$  is the set of edges of  $G$ .

**Definition 1:** A *cluster* of  $G$  is a set of two or more nodes of  $G$  such that one of the nodes is directly connected to each of the others. The nodes in a cluster are *members* of the cluster. One of the cluster member nodes that is directly connected to all the others is distinguished as the *representative* of the cluster.

**Fact 1:** It is possible to decompose any connected graph into a set of clusters such that every node is contained in a cluster. (A straightforward inductive argument shows this.)

**Fact 2:** Any cluster decomposition of a connected graph  $G$  contains no more than  $\lceil |G|/2 \rceil$  clusters. (This follows from the fact that every cluster has at least two nodes.)

**Definition 2:** Let  $G = \langle V, E \rangle$  be a connected graph and let there be a cluster decomposition of  $G$ . The *cluster-induced graph of  $G$  with respect to this decomposition* is a graph  $G' = \langle V', E' \rangle$  where  $V'$  is the set of clusters of  $G$ , and  $E'$  has an edge from  $C_1$  and  $C_2$  (where  $C_1$  and  $C_2$  members of  $V'$ ) if there is an edge of  $G$  that connects a node of  $C_1$  with a node of  $C_2$  in  $G$ .

This technique of constructing induced graphs may be used to form a hierarchy of cluster levels.

**Fact 3:** If  $G$  is a connected graph, it is possible to define a sequence of graphs  $G = G_0, G_1, G_2, \dots, G_n$ , where  $G_n$  is the trivial graph consisting of a single node, and  $G_{i+1}$  is a cluster-induced graph of  $G_i$  for each  $i = 0, \dots, n-1$ , where the length of the sequence,  $n$ , is no more than  $\log_2 |G|$ .

**Remark:** As members of  $V'$ ,  $C_1$  and  $C_2$  (in Definition 2) are nodes in  $G'$ , but they are also clusters of nodes in  $G$ . It is convenient to *name* the clusters by giving them the same name as its representative node. For example, if  $G$  is the graph of the network shown in Figure 9 of section 3.3.3, then Figure 10 shows the graph  $G' = G_1$ , the induced graph of  $G$ . The node “Z” of the graph  $G$  is also the name of a cluster in  $G$ , or equivalently, a node in the induced graph  $G_1$ , shown in Figure 10.

**Definition 3:** A node is a *level  $i$  cluster representative* for  $G$  if it is a node in  $G_i$ . Note that a level 0 representative is simply a node of  $G$ , and a level 1 cluster representative is a cluster representative as defined in Definition 1 above.

Finally, we can obtain network locations for each node can be obtained from the cluster levels:

**Definition 4:** The *network location* or *full-cluster address* of a node  $X$  in a network  $G$  is  $[X_n, \dots, X_0]$  where  $X_i$  is the level  $i$  cluster representative of  $X$  for  $i = 0, \dots, n$ .

## Clustered Route Advertisement

The advertisement system provides a mechanism that permits network nodes to determine the cost of sending packets to other nodes. To improve scalability, advertisements propagate selectively, and nodes do not advertise to all other nodes, just nodes within certain clusters.

The advertisement system enables nodes to inform each other about distances and costs in the network. Locally, each node develops knowledge about how far other nodes are and how costly it is to reach them using available links. Each node stores this information in a *routing table*, which contains an entry for each destination-link pair (for each destination about which it has received advertisements). For a given destination  $D$  and link  $L$ , the  $D$ - $L$  entry in the routing table contains the cost of reaching node  $D$  via link  $L$ .

Since the information in advertisements becomes outdated quickly as network loads and topologies change, advertisements propagate from each node at fixed periodic intervals. Recipients of advertisements obtain a new snapshot of the nearby topology with each new set of advertisements. To prevent anomalies such as routing loops and the *counting to infinity* problem, the advertisement system keeps track of the most current and consistent set of advertisements, referred to as an *advertisement edition*.

The goal of the routing layer is enabling nodes to make nearly-optimal routing decisions, i.e. the same decisions it would make if it had global network knowledge. If a node had knowledge of the full network topology, it could use Dijkstra's "shortest paths" algorithm to determine optimal routes in the network. In the CoCo scheme, no node has complete network information.

However, by exchanging information with neighboring nodes, and by working in conjunction with the clustering system, the protocol finds routes that are close to optimal while exchanging far fewer messages than would be required if each node sent advertisements to all other nodes. The number of messages is limited by *rank-based advertisement propagation*, the principle that a node's advertisements propagate throughout the network based on its rank. The higher a node's rank, the more widely it is advertised through the network. This limits advertisement propagation and helps control protocol overhead.

## Location-based Routing

In general, a node's routing table does not contain an entry for every device in the network. However, using the notion of network locations, a node  $S$  can route data to a destination node  $D$  effectively *even if the destination node  $D$  does not appear in its routing table*. If device  $S$  wants to send data to a device  $D$  with network location  $[D_n, \dots, D_0]$ , it uses the algorithm in **Error!**

**Reference source not found..**

```

// S tries components of D's location, starting with the closest
for i = 0, ..., n
{
  if (Di occurs in the routing table)
  {
    L = best link to send to Di according to the table;
    send the packet over link L;
    break;
  }
}

```

Figure 6: How Node D Chooses a Route to S

The higher the value of the subscript  $i$ , the more widely advertisements for  $D_i$  propagate through the network; therefore, the more likely S will have received one and have an entry in its routing table for  $D_i$ . Once S sends a packet to  $D_i$  there is a high probability that  $D_0$  occurs in  $D_i$ 's routing table. If not,  $D_i$  applies the same location-based technique to obtain a route to  $D_j$  (for some  $j < i$ ). If  $N$  is the number of devices in the network, then  $n < \log_2 N$  bounds the number of potential re-routings.

In practice, further optimizations are possible. Consider the scenario above in which S sends a packet P to  $D_i$  (because  $D_i$  is the closest component of D to D itself that occurs in S's routing table). As the packet P moves toward  $D_i$  it passes through nodes along the path from S to  $D_i$  that will likely have better information about reaching D. This is a consequence of rank-based advertisement propagation: the closer a node is to D, the more likely it has received advertisements from D.

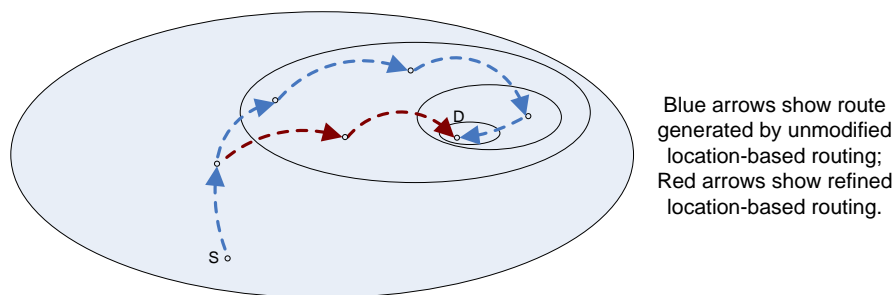


Figure 7: Refinement in Location-based Routing

As the packet gets closer to D, the more refined the information in the routing tables of nodes on its path becomes. Hence the actual path traversed by the packet from S to D may be much shorter than the path that passes through the components of D's location:  $S \rightarrow D_i \rightarrow D_{i-1} \rightarrow \dots \rightarrow D_0 = D$  implied by Figure 6. The blue arrows in Figure 7 indicate the route based on pure location-based routing; the red arrows indicate the route based on refined location-based routing.

## Circuit Layer

In the CoCo Protocol, a *circuit* is a communication path over which data moves from one device to another. The circuit layer is the first layer in the CoCo Protocol that supports end-to-end communication, which may be encrypted on a per-circuit basis. This represents a separate application of encryption from that used at the link layer discussed in section 3.2. The circuit layer manages the creation, maintenance, and destruction of circuits. The circuit layer also manages handoffs—adjustments to the circuit path made necessary by CoCo devices changing position.

A circuit consists of *legs*, where each leg uses one link. A circuit may be in any of three states:

- *C: Closed* (nonexistent)
- *O: Opening* (in the process of being created)
- *R: Ready* (ready for data to traverse it)

Circuits are unidirectional: the existence of a circuit from A to B does not imply the existence of a circuit from B to A. If B wants to send data to A, it must establish a new circuit from B to A, separate from the circuit from A to B. Such a circuit from B to A may not follow the reverse path of the circuit from A to B because some network links may be slower in one direction than the other.

## Circuit Establishment

When a node S wants to communicate with a node D it consults the routing layer to determine the best link for packets destined to D, and sends a circuit establishment control packet over that link. This packet contains the following data:

- destination
- QoS requirements
- Circuit ID (see Section 3.4.2)

When a node A receives a circuit establishment packet, it checks to see if it is the intended destination. If not, node A forwards the establishment packet to one of its neighbors and changes its state from *C* to *O*. It determines the link over which to forward the message by consulting the routing table. If node A is the final destination (i.e. A and D are the same node), then A sends an acknowledgement packet back toward the original initiator node, S. Each intermediate node, upon receipt of an acknowledgement packet, similarly sends an acknowledgement packet along the circuit backward toward S. When a node receives an acknowledgement packet, the circuit state changes from *O* to *R*. When the original initiator node D finally receives an acknowledgement packet and changes its state to *R*, the circuit is fully established and ready for S to begin sending data packets to D.

## Circuit Tables

Each node may be a part of several circuits. The circuit layer maintains a *circuit table*, an internal data structure that enables it to associate inbound links with outbound links, for each active circuit passing through a node.

The Circuit ID (CID) is a number that associates packets arriving over a particular link with a particular circuit. The CIDs associated with different legs of a single circuit may be different. For example, if a packet containing  $CID = v_1$  arrives at node N from link  $l_1$ , the circuit layer consults its circuit table to determine that the packet should be forwarded along, say, link  $l_2$  with  $CID = v_2$ . If node N is the packet's final destination, then the circuit layer forwards the data to a user application process specified by the endpoint address (similar to a TCP port) that appears in the packet header.

The circuit layer uses the circuit table to send control packets as well as data. Control packets for opening and closing circuits move in the *forward direction*, i.e. the direction of data. Control packets for acknowledgements and resetting the circuit, if necessary, are sent in the reverse direction. The circuit table contains sufficient information to enable this.

## Circuit Layer Multipath Support

It is possible for circuits to maintain multiple paths between any pair of nodes along the circuit, so in the general case, a circuit is represented locally at each node by a set of incoming and a set of outgoing legs. This feature permits greater transmission options. Since some links have different performance characteristics (bandwidth, latency, etc), the circuit layer may be able to satisfy user QoS requirements more easily when it has more links from which to choose. To illustrate, consider a circuit from node W to node Z in the network topology of Figure 8.

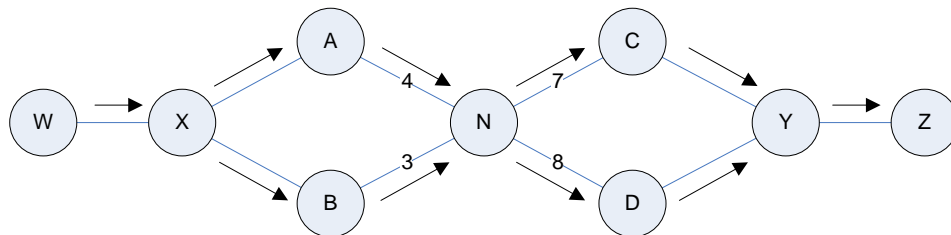


Figure 8: A Multi-path Circuit

At the node N, the circuit table includes the information that any data inbound from A with  $CID = 4$  or from B with  $CID = 3$  must be forwarded to C with  $CID = 7$  or to D with  $CID = 8$ .

Multipath support for circuits enables a natural method for circuit handoffs. If a node that is part of a circuit moves from one geographic location to another, the links between it and the other nodes in the circuit may become weaker than links to other nearby nodes. When this happens, the circuit layer

anticipates the links drop and adds legs to the circuit. Initially, they may be redundant, but they can effectively replace links that break due to the geographical movement of a node. This enables a seamless handoff not only from one node to another, but from a link that uses one transport mechanism to another link that uses a different transport mechanism between the same two nodes. For example, a pair of nodes A and B may have a WiFi link and a cellular link, and one may strengthen as the other weakens. Multipath support also enables bandwidth aggregation.

Figure 9 illustrates a network with a circuit established from node W to node Z. The series of illustrations in Figure 9 show the effect on this circuit as node N moves. In Figure 9a, N has good reception to nodes A and C and none to B and D, so the circuit goes through A and C. As Node N moves, as Figure 9b illustrates, N starts to receive a signal from nodes B and D, but the signal is too weak for the circuit to add legs through B and D. When N is equidistant from A, B, C, and D (see Figure 9c), the signals to these nodes are all strong enough for links to form. These links enable new circuit legs to form from B to N and from N to C, which enhances the bandwidth available through N. As N moves away from A and C toward B and D, the signals to A and C weaken and eventually drop, leaving just the path through B and D (Figure 9e). The circuit from W to Z continually adjusts to make use of available links. As links form or break, the circuit layer updates the circuit tables in the affected nodes.

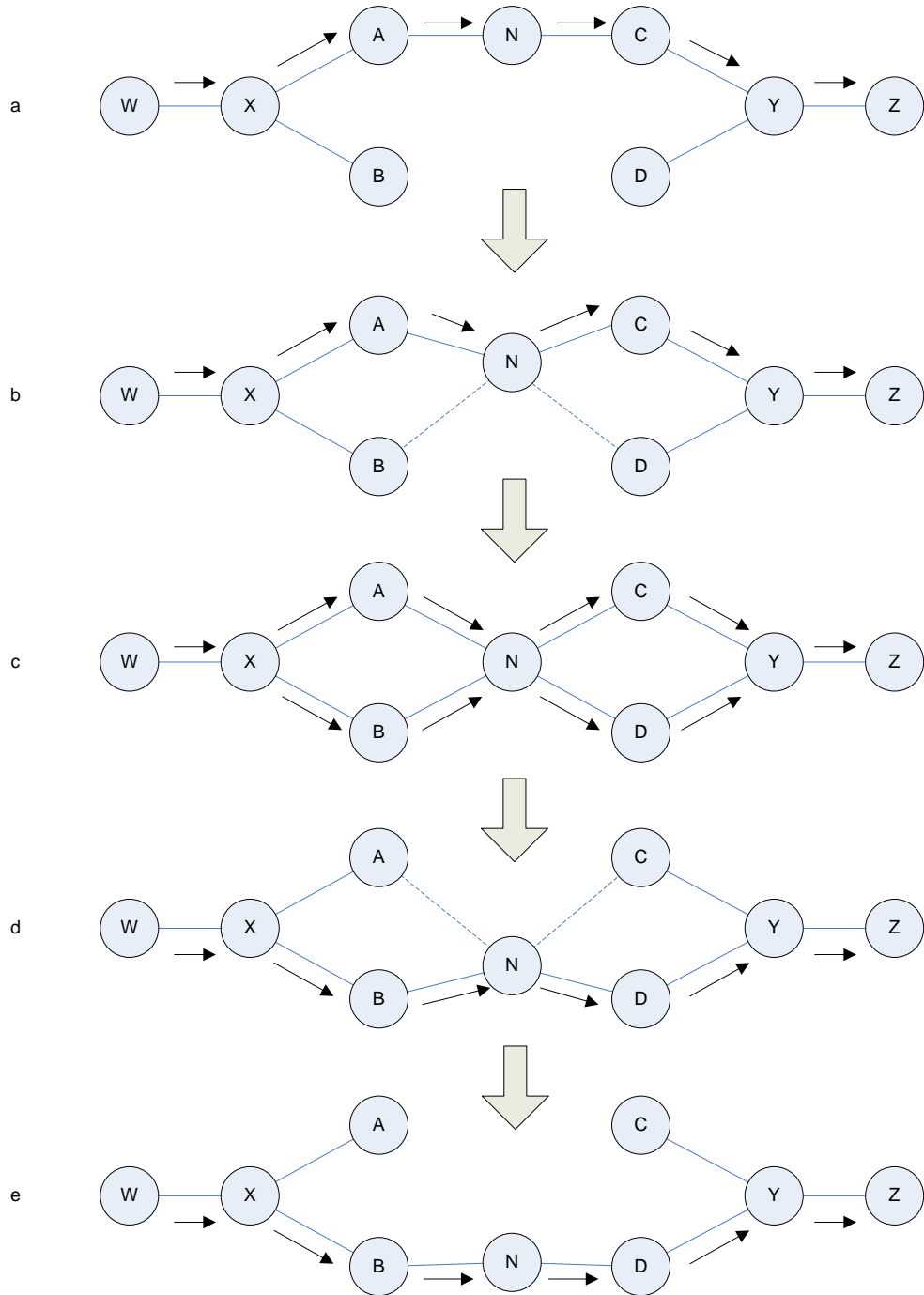


Figure 9: Circuit Layer Handoff Illustration

## Circuit Layer Multicast Support

The circuit layer protocol includes multi-transport datagram multicast. Multicast support enables data sent from a single source to multiple destinations to be transmitted non-redundantly—hence more efficiently—in the sense that only one copy of the data packets is sent across shared links from the source to the destinations. For example, if A wants to send a packet

to each of B and C in Figure 10, only one copy of the packet is sent along the link from A to X.

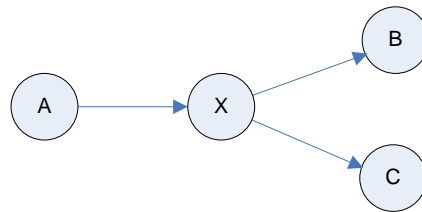


Figure 10: Circuit Layer Support for Multicast

For messages sent to many recipients along paths that share a significant number of links, multicast support represents a substantial reduction in bandwidth utilization.

## Identity Layer

The concepts of *name* and *location* as they apply to CoCo networks were introduced at the beginning of this document. As DNS maps names to IP locations, so does the CoCo Identity layer. Since devices may appear to change location on a regular basis, especially in ad-hoc routing scenarios, the process of name resolution must survive catastrophic network events.

Our peer-to-peer identity management system provides a temporary replacement for DNS. Such a system necessitates a level of cryptographic certainty that responses can be trusted and queries should be processed, so CoCo uses X.509-encoded, chain-signed, PKCS-compatible certificates to match a public key to a DNS-compatible domain name. For each certificate, the Identity layer instantiates one security *role*.

## Distributed Name Resolution

The naming system is completely decentralized and distributed. It self-generates when a network first forms and it self-adjusts when the network topology changes. DNS, by comparison, requires human intervention to update server IP addresses and much longer delays for such changes to propagate through the network. Names are hierarchically structured ASCII text strings that cannot be forged. Each device may be assigned a name when it is originally provisioned. It is possible to delegate the authority to assign names. For example, the city of Seattle may delegate to the police commissioner the authority to distribute names for the network devices used by members of the Seattle Police department. These operations are completely external to the CoCo network system and are wholly driven by the configuration of the certificate authority. This provides the maximum flexibility in defining security relationships that can be automatically enforced.

## Hierarchical Structure of the Namespace

The naming system is hierarchical. For example the name `smith.police.seattle.wa.us` implies five logical tiers of the system as represented in Figure 11.

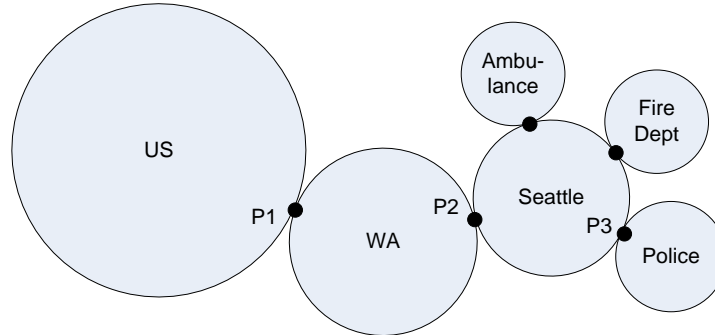


Figure 11: Exemplary hierarchy of identity

The expanded hierarchical view of the name space in Figure 11 illustrates the dominance of local network traffic as it is far more likely that nodes of similar identity will be proportionally more likely to communicate than nodes of less similar identities. It is reasonable to expect, for example, that most of the traffic to and from the device with name `smith.police.seattle.wa.us` would involve devices with names of the form `*.police.seattle.wa.us` more than with devices with names of the form `*.police.beijing.china`.

## Naming Convergence

The convergence algorithm uses a series of *registration* messages among selected nodes in the network. A node X that wishes to join the network initially detects the presence of another node Y and establishes a link to it as described in the Routing Layer section. To integrate itself into the naming system, node X sends a registration message to introduce itself to Y. Node Y then computes the tree-relation of the name of node X to the name of node Y, with results such as “parent” or “child” resulting in local tree reorganization, “descendent” or “distant” resulting in message forwarding along the existing tree, or “sibling” which instantiates or expands a multicast group.

This constructs a spanning tree of names where any tree element may also represent a multicast group. Aside from the implied state maintenance cost, this algorithm is considered to be academically understood and intuitive.

## Address Translation Layer

The elements presented thus far demonstrate the CoCo Protocol’s ability to make use of existing Layer 2 and Layer 3 transports to construct a peer-to-peer topology with stateful routing to affect a data-over-voice transmission system among devices with certified security roles. While this would enable custom application development, CoCo’s perspective dictates that no feature

should necessitate changes to common Internet applications such as the web browser. This means that all control and signaling must happen through an IP-compatible interface.

## IP Compatibility

Today's host operating systems are fairly standard in their reliance upon IP sockets, which in turn requires that expansions to the system be delivered in the form of network interfaces. An example from the COTS market would be the common VPN or Wi-Fi management software which installs a new network driver into Microsoft Windows or Debian Linux.

The most common message exchanges are DNS name resolution, TCP or UDP packet routing, ICMP signaling, and IGMP group management. Our protocol stack is capped with a translation module to exchange instructions between the host IP stack and the multi-transport, multicast logical view of the CoCo network. So when a network circuit disconnects, we may generate a messages such as *TCP reset* or *ICMP host unreachable* to effectively instruct the IP stack. This is how Internet Explorer and IIS work together perfectly over CoCo even on a pair of laptops in a desert with no DNS implementation.

## Network Address Translation (NAT)

NAT is most commonly used to proxy multiple machines on a private network through a single gateway device so that many users can share one publicly routable IP address. CoCo uses this exact technology in a reversed configuration to proxy the entire CoCo network through one private IP address range. In other words, my machine may alias the name `smith.police.seattle.wa.us` to a private IP address, say `10.0.0.2`, so that the system is independent of any IP assignment authority. This avoids substantial responsibility at provisioning time and also avoids the need for on-site configuration management servers.