

# Understanding Expected Delay (ED)

Document ID: 29522

---

## Introduction

### Prerequisites

- Requirements

- Components Used

- Conventions

### Background Information

### Troubleshoot Expected Delay

### Extrapolation

- Extrapolation Example

### Set Expected Delay Traces

### Other Tools to Troubleshoot Expected Delay

- Syntax Example

### Related Information

---

## Introduction

This document lists some common problems related to Expected Delay (ED), and explains how to calculate ED, where the data comes from, and how to troubleshoot issues.

## Prerequisites

### Requirements

Cisco recommends that you have knowledge of these topics:

- Cisco Intelligent Contact Management (ICM) configuration and scripting
- Automatic Call Distribution (ACD) concepts and call flows, as related to Cisco ICM
- The Cisco ICM rttest Utility
- How to Use the Dumplog Utility or InspectLog Viewing Utility
- OPCtest Utility
- Procmon Utility
- How the CallRouter uses extrapolation when it performs calculations

### Components Used

The information in this document is based on these software and hardware versions:

- Cisco ICM 4.6.2 and later

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

### Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

# Background Information

ED is a metric used in Cisco ICM, Cisco Network Applications Manager (NAM), and IP Contact Center (IPCC) environments.

In general terms, ED is the predicted delay (in seconds) for any new call added to a queue for a Service. ED is valid only if no agents are available.

**Note:** If agents are available, ED is zero.

Minimum Expected Delay (MED) is a standard selection rule available in the **Select** and **Route Select** nodes of the Script Editor. If you select from multiple services, and use the standard MED rule, the CallRouter selects the service with the smallest value for MED (the minimum).

In order to fully understand ED, you must know how ED is calculated.

**Note:** ED is a service-only calculation.

You cannot route with MED to a set of skill groups. Here is the standard ED formula:

$$((\text{CallsQNow} + 1) * \text{AHTto5}) / \text{Max} (\text{Agents Talking [OR] Ready})$$

- **CallsQNow** is a count of the current calls in queue for the service at the peripheral.
- **+1** is used to indicate a call that can be potentially added to the queue.
- **AHTto5** is defined as the average handle time (in seconds) for calls to the service during the current five-minute interval. AHTto5 is a rolling five-minute average (from now, and for the most recent five minutes), and is calculated in real-time. The value for AHT is calculated as:

$$\text{HandleTimeTo5} / \text{CallsHandledTo5}$$

- **HandleTime** is tracked only for inbound ACD calls that are counted as handled for the service. HandleTime refers to the total time spent on a call. Therefore, HandleTime is the total call duration, from the time the agent answered the call to the time the agent completed the after-call work. HandleTime includes any TalkTime, HoldTime, and WorkTime associated with the call (from Termination\_Call\_Detail). The AvgHandleTime value is updated in the database when the agent completes all the after-call work associated with the call.

**Note:** If there were no inbound ACD calls handled for the service during the most recent five-minute interval, Cisco ICM uses a default AHT value of 120 seconds in the ED formula. You cannot configure this default AHT value. It is hard-coded in the `router.exe` application.

In the denominator, the CallRouter uses either the AgentsTalking value, or the AgentsReady value (whichever value is currently higher).

- The value for AgentsTalking is the number of service agents currently in the talking state. The AgentsTalking value includes all skill groups in the service (as defined in the Service\_Member).
- The AgentsReady value comes from the Skill\_Group\_Real\_Time table, and includes agents in the Ready state. Ready is a state in which an agent is logged on to the system, and is either on a call currently, or involved in after-call work, or is available to handle a new call. As mentioned previously, ED assumes that no agents are available. The AgentsReady value includes only those agents in skill groups defined as *primary* in the Service\_Member.

**Note:** Some ACDs support agents in multiple subskills, with different priorities. The CallRouter considers AgentsReady, and only includes those agents who are members of subskill number ONE (1).

# Troubleshoot Expected Delay

When you understand how ED is calculated, you can troubleshoot situations where the ED formula results in unexpected values. Many times, you can trace a problem with ED to a mismatch in Cisco ICM and ACD configurations, because the problem pertains to a Peripheral Service. Ensure that the Service and Skill Group peripheral numbers are correct, and that the Service\_Member information is accurate. Ensure that agents are logged into the member skill groups. If you use subskills, ensure that the agents are logged into the subskill number one (1).

If the configuration is accurate, enable specific traces in order to ascertain the problem.

## Extrapolation

Here is a brief explanation of the extrapolation mechanism of the router. This section explains why extrapolation is necessary and how it is implemented.

### Extrapolation Example

Assume that a simple routing script attempts to load balance calls based only on the number of calls in queue, and sends the call to the site with the fewest calls.

**Note:** Although this example refers to calls in queue, the same mechanism is used for a number of other variables, listed later in the document.

1. A call arrives.
2. The router picks a site, and sends the call.
3. The network delivers the call.
4. The ACD sees the call arrive, and runs an internal script that places the call in queue.
5. Cisco ICM (through the PIM and OPC) notices the call and the change in queue statistics.
6. Cisco ICM reports back to the router, where the number of calls in queue is updated.

All of this takes time to happen. It can take seven seconds for all these steps to occur. For those seven seconds, the router still thinks the number of calls in queue is the original value. If the router is given a new call to route, the router still thinks the same site is the best site. In a high volume application, you can easily send dozens of calls to the site before you finally receive an updated queue count from the PG. At that point, some other site suddenly looks much better, and the router sends all calls to that site. The phenomenon is called fire hose routing .

This is simply an example. The amount of time depends on the network, ACDs, or VRUs involved. The router has limited information to resolve this issue. In particular, there is no way for the router to match incoming data from the PG with actual calls that are routed. Therefore there is no way to know, for example, which calls are included in the calls in queue metric when the PG reports the queue count.

The extrapolation mechanism in the router is a solution implemented in Cisco ICM. The mechanism is used to try to estimate the real value. Here is how extrapolation works for a variable like CallsQueueNow for a service:

Internally, CallsQueueNow is managed in two parts:

- CallsQueueNow base value, which is the value last reported by the PG.
- CallsQueueNow adjustment, which is managed by the router.

When a routing script references CallsQueueNow, it sees the sum of the base value and the adjustment. When CallsQueueNow is sent in the real time feed to the AW, only the base value is sent. In order to manage the adjustment, the router adds 1 when the call is routed to the service, and then sets a timer. The default value for the timer is 10 seconds. When the timer expires, the router subtracts 1 from the adjustment.

Here is an example with actual numbers:

Assume that there are 3 calls in queue:

1. At the start, base=3, adjustment=0
2. A call arrives, and is routed to the service, base=3, adjustment=1. Other calls routed at this point see 3+1=4 calls in queue.
3. Seven seconds later, the PG reports there are 4 calls in queue. Now base=4, adjustment=1 (still). Calls routed at this point see an overestimated value of 5 calls in queue.
4. Three seconds later, the 10-second extrapolation timer expires. Now base=4, adjustment=0.

This example indicates an overestimation of the number of calls in queue.

Similar mechanisms are used on a number of routing parameters. This table lists the variables that are extrapolated:

Object	Fields	Direction
Service	CallsQNow	Up
	ExpectedDelay	Up
	CallsInProgress	Up
	CallsInNow	Up
Skill Group	AgentsAvailable	Down
NetworkTrunkGroup	TrunksIdle	Down
	CallsInNow	Up

The direction column indicates the direction in which the adjustment is made [+1 (Up) or <sup>TM</sup> (Down)]. An extrapolation mechanism is also used to manage agents.

In particular, the LongestAvailableAgent variable is managed through a mechanism that is entirely different from what is described here. The router receives status on individual agents from the PG. Internally, it maintains a list of all available agents, ordered by the time when the agent becomes available.

When an agent is selected (for example in LAA), the router marks the agent at the head of the list as temporarily unavailable for 10 seconds. During this time, the PG ignores the state report, and the router assumes that the agent is unavailable. After that time, the agent state reverts to whatever the PG last reported. This mechanism allows the router to account for the use of specific agents, and enables recovery if the ACD happens to send a call to the wrong agent. This kind of routing can be more precise than the other metrics. This is because no adjustments are made as long as the ACD sends the calls to the agents that the router guesses.

Sometimes, there can be a confusion about the behavior of AgentsAvailable and LongestAvailable. AgentsAvailable is adjusted by the up/down algorithm, and can underestimate the number of agents available. LongestAvailable is computed independently from the available agent list. LongestAvailable can show an agent available even though AgentsAvailable indicates zero. Therefore, LongestAvailable is more accurate, as mentioned earlier.

# Set Expected Delay Traces

Expected Delay traces display values that are extrapolated, and you can implement the traces through **rttest**.

```
trace_ed N
```

where N is the SkillTargetID of a service. This command turns on the trace.

```
trace_ed N /off
```

This command turns off the trace.

When you enable this trace, the CallRouter puts debug level log entries in the console window and in the .EMS log file. Use dumplog or InspectLog Viewing Utility to view the log file output. The router prints this message:

```
ED RR NAME(ID) xNN B=(qNN rNN tNN aNN hNN eNN) E=(qNN rNN tNN aNN hNN eNN)
```

RR represents the reason for the trace. Here are the various code descriptions:

Code	Description
T+	Trace is turned on.
T-	Trace is turned off.
E+	An extrapolation is started (this is caused when a call is routed).
E-	An extrapolation ends (the 10-second timeout).
SK	Updated because a skill group variable changed (the PG reports the change).
SV	Updated because a service variable changed (the PG reports the change).

- **NAME (ID)** represents the name and ID of the service.
- **XNN** is the number of extrapolations in progress. This is the number of calls in the last 10 seconds.

Here are some code descriptions:

Code	Description
QNN	Calls in queue.
Rnn	Agents ready.
Tnn	Agents talking.
Ann	Agents available.
Hnn	Average handle time to 5.
Enn	Expected delay.

There are two sets of these variables:

- **B=()** set is the base set of all the variables, as reported by the PG, and ED calculated from them.
- **E()** set is the extrapolated set, based on recently routed calls.

## Other Tools to Troubleshoot Expected Delay

You can use the Display RealTime Data feature of the Script Editor to troubleshoot MED. It is important to know that the data displayed in Script Editor can be as old as fifteen seconds or more, and often only displays base values, rather than extrapolated values.

Look at the data in real time to troubleshoot ED. For this, use the **dump\_vars** command from within `rttest`, to view the various values and variables that the CallRouter knows.

```
rttest: dump_vars /?
```

**Note:** The values that are listed can be extrapolated.

### Syntax Example

In `rttest`, run:

```
dump_vars /service <Service.SkillTargetID>
```

or

```
dump_vars /group <Skill_Group.SkillTargetID>
```

You can determine the **SkillTargetID** through ISQL/W or the Quick Query feature found in the Schema Help program.

If you enter a proper value for the Service or Skill Group SkillTargetID, **rttest** displays a list of the variable names (for example, AgentsAvailable and AgentsReady) and a column with the value of each variable. Usually, the value is a positive integer, and self-explanatory. -1 indicates that the value is undefined.

When you troubleshoot, compare the values seen in **rttest**, **dump\_vars** with information available from the ACD. When you compare data, look for a possible irregularity that can be the cause of the problem.

Some Cisco Customer Support Engineers (CSEs) have also had success with the **watch** command in `rttest`. The **watch** command enables you to evaluate any applicable expression. The **watch** command is most useful to troubleshoot custom formulas (for example, custom ExpectedDelay calculations). If you change the expression value(s), the CallRouter immediately includes an entry in the router process window (and in the `.ems` file) with the current value.

Here is how you must issue the **watch** command:

```
rttest: watch <expression>
```

where:

- The `expression` is any valid expression, for example:

```
rttest: watch Service.Boston_Aspect.Support.AgentsReady
Watch 0 added.
```

- You can remove the watch through the **/delete** switch, for example:

```
rttest: watch 0 /delete
```

OPCTest and Procmon also have various sub-routines that allow you to list agents and calls. Cross-reference these values with what you know about the ACD, and the CallRouter. Look for a possible irregularity that can be the cause of the problem.

If you recently installed Cisco ICM, and you bring up a new service for the first time, the MED can be different from what you expect. Many times, the MED is different because of one of these reasons:

- Effects of extrapolation.
- No calls are handled (default is 120 seconds for AHT, and cannot be expected).
- Few calls are in progress or in queue.

ED is most accurate when there are many items to average. When more agents are available in the member skill groups, and more calls are handled, the MED results are better.

---

## Related Information

- [How to Use the Dumplog Utility](#)
- [InspectLog Viewing Utility](#)
- [The Cisco ICM rttest Utility](#)
- [Using the OPCTest Command Line Utility](#)
- [Using Remote Process Monitor Console \(Procmon\)](#)
- [Technical Support & Documentation – Cisco Systems](#)

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Apr 25, 2005

Document ID: 29522

---