

Troubleshooting and best practices in Claude Dispatch

Complex, multi-step Dispatch tasks are only successful about 50% of the time.

Only 50% of Dispatch's complex, multi-step tasks are successful.

Quick summary : In lesson 6, we learned about Dispatch's security architecture—three layers of protection that keep your data safe. But security is useless if your mission never gets done. Let's talk about reliability.

This is a number we've mentioned many times: Complex, multi-step Dispatch missions only succeed about 50% of the time. MacStories has verified this. Reddit users have confirmed it. And Anthropic doesn't disagree.

50%, that's too much of a gamble.

That sounds bad. And for complex tasks, that's somewhat true. But the point is: You can significantly improve your success rate by understanding why the error occurs and adjusting how you use Claude Dispatch .

Simple tasks (reading files, summarizing, searching) succeed 80-90% of the time. The gap between 50% and 90% isn't due to Dispatch errors – it's due to task design. And task design is something you can control.

Why Dispatch failed: The root cause.

Reason 1: Your Mac goes into Sleep mode (40% of all errors)

This is the main reason. Your Mac goes into Sleep mode, the Cowork process pauses, the connection is lost, and your task fails. You won't find out until you try to send the next task.

How to fix it:

```
caffeinate -d &
```

Run this command in Terminal before you leave. It keeps your Mac running until you close the Terminal window. The -d flag also prevents the screen from going into Sleep mode.

A better solution : Change System Settings > Energy to prevent Sleep mode when plugged in. This setting remains even if you close the Terminal window and restart the application.

Reason 2: Unstable network (25% of errors)

Dispatch requires a stable connection between both devices and the relay server. If your Mac's Wi-Fi is interrupted for 30 seconds or your phone loses signal in an elevator, the mission may fail.

Fix :

1. Use Ethernet on your Mac whenever possible.
2. Do not rely on Dispatch in areas with weak phone signals.
3. If you know you'll be in an area with weak signal, send the task before you enter the area of poor signal.

Reason 3: Task complexity (20% of errors)

Tasks requiring multiple sequential steps, connector interactions, and lengthy processing times are more likely to fail. This isn't because Dispatch can't handle them—but because each step is a potential point of error, and the probability of failure increases exponentially.

A 5-step task, where each step has a 90% success rate: $0.9 \times 0.9 \times 0.9 \times 0.9 \times 0.9 = 59\%$ overall success rate.

The same task is divided into 5 individual tasks: Each task has a 90% success rate independently, and you can retry individual errors.

Solution : Break it down. Always break it down.

Reason 4: Connector waiting time (10% of errors)

Some connectors have API timeout windows that conflict with Dispatch latency. A task that takes 2 seconds on the local machine might take 8 seconds via Dispatch (with added encryption and forwarding), exceeding the connector's timeout.

Troubleshooting : Check connectors before using them. If a connector consistently times out, switch to file system-based workflows.

Reason 5: All other causes (5% of errors)

Cowork updates, temporary server issues, exceptions in the relay system, or truly new bugs. This is a preview version of the software for research. Some bugs simply happen.

Solution : Try again. If it fails twice, try a different approach or wait.

The fragmentation strategy

This is the most effective technique to improve Dispatch's credibility. It's not complicated, but it requires a shift in mindset.

Instead of doing this:

Xem xét PR trong ~/projects/app, kiểm tra các vấn đề báo lỗi, sửa lỗi nếu có TypeScript nào, cập nhật các bài kiểm tra và cam kết các thay đổi

Please submit each request one at a time:

Cho tôi biết những file nào đã thay đổi trong PR mới nhất trong ~/projects/app

Wait for a response. Then:

Chạy src/auth/login.ts từ PR và ánh xạ dữ liệu lỗi về vấn đề báo lỗi nào

Wait. Then:

Có lỗi TypeScript nào trong các file đã thay đổi không?

And so it continues. Each task takes 10-30 seconds. The total time is similar to the method of sending only one request. But the success rate increases from ~50% to ~85%.

Quick test : Take your most complex Dispatch task and break it down into the smallest possible steps. How many steps do you get? If it's more than 5 steps, consider whether you really need to perform all of those steps remotely, or whether some steps can wait until you're sitting at your desk.

Diagnostic flowchart

When Dispatch fails, follow this checklist in order:

Tác vụ thất bại? Bị lỗi Dispatch có đang hoạt động trên nền tảng của bạn không? **KHÔNG** ? Kiểm tra máy Mac của bạn có đang hoạt động không? **KHÔNG** ? Kiểm tra máy Mac, ghép nối lại? **CÓ** ? Kiểm tra kết nối Internet của máy Mac, ghép nối lại? **CÓ** ? Kiểm tra đang hoạt động? Gửi email bài kiểm tra ngắn gọn: "Liệt kê các file trên Desktop của tôi" **THÀNH CÔNG** ? Nhiệm vụ ban đầu của bạn quá phức tạp? Chia nhỏ nhiệm vụ thành các bước nhỏ hơn và thử lại? **THẬT BÀI** ? Kiểm tra đang hoạt động nhưng không nhận được? Chờ 60 giây và thử lại? **THÀNH CÔNG** ? Sửa lỗi tạm thời, tiếp tục? **THẬT BÀI** ? Ghép nối lại? **THỰC HIỆN**

This diagram solves 90% of Dispatch issues. Print it out, take a screenshot, and memorize it. The steps take less than 2 minutes.

Best practices for reliable Dispatch operation.

This is a summary guide from pioneering users who have achieved a success rate of over 80%:

1. Prevent your Mac from going into Sleep mode (Non-negotiable)

```
caffeinate -d &
```

Alternatively, adjust the Energy setting. This alone can fix 40% of all the problems.

2. Use Ethernet whenever possible.

Wi-Fi works well in most cases. Ethernet works well in all cases. If you plan to use Dispatch for an extended period, plug in a cable.

3. One task, one message.

Never combine multiple actions into a single message, even if it's appealing or saves typing. Simplifying the process doesn't benefit you.

4. Verify after each recording.

After performing any file modification task, verify the results:

```
Cho tôi xem nội dung của ~/Documents/report.md
```

This helps detect incomplete recording operations and gives you a chance to correct them before continuing.

5. Always use the full path.

```
??c ~/projects/my-app/src/config.ts
```

Not required:

```
??c file của mình
```

Ambiguity leads to errors. Be clear!

6. Begin with reading operations.

When starting a Dispatch session, begin with a read operation to verify that the connection is active. Don't start with a complex write operation and only discover the connection is broken after you've entered a long prompt.

7. Save the important results to a file.

```
Lưu các kết quả vào ~/Documents/dispatch-output/analysis.md
```

Phone screens are small and session memory is only temporary, while files exist permanently.

8. Have a contingency plan.

For each Dispatch process, know what you'll do if it fails. Sometimes the backup plan is "try again." Sometimes it's "this can wait until I sit down at my desk." Having a backup plan will help avoid frustration.

Common error messages and how to fix them.

Error/Symptom	Possible causes	Fix
"Connection lost"	The Mac has gone into Sleep mode or lost its internet connection.	Start your Mac, check your internet connection, and then reconnect.

Error/Symptom	Possible causes	Fix
No response (currently recording)	The task has timed out.	Simplify the task, try again.
"Failed to access file"	Incorrect path or file does not exist	Double-check the full path.
The approval notification never arrived.	Phone number blocked	Check the notification settings for the Claude app.
Partial (interrupted) feedback	The limit or timeout of the context window.	Request less in a message.
Connector error	Connection timeout error or authentication error	Test the connector separately, using a file-based backup solution.
"Session expired"	Long periods of leisure time	Reconnect using the new QR code.

Reliability

A candid look at Dispatch's current position and future direction:

March 2026 (present):

1. Simple tasks: Success rate 80-90%
2. Complex tasks: Success rate ~50%
3. Overall breakdown: Success rate ~80%

Proposed course of action:

1. Anthropic will improve rapidly.
2. Compatibility with connectors will be improved.
3. Network processing will be more powerful.
4. Background execution (Mac's ability to withstand Sleep mode) is in the development roadmap.

Dispatch is like the iPhone in 2007 – you can see its potential even when it's still imperfect. Those who learn it now will become experts when it's perfected.

Exercise: Create your troubleshooting guide.

Create a personalized guide for your most common Dispatch scenarios:

1. List your top 5 Dispatch tasks.
2. For each task, note: what typically goes wrong, how to fix it, and your contingency plan.
3. Add a diagnostic flowchart from this lesson.
4. Save it on your phone (Notes app, screenshot, whatever is handy).

This guide transforms 5-minute troubleshooting sessions into solutions that take just 30 seconds.

Key points to remember

1. The error rate of approximately 50% for complex tasks drops to around 80-90% when you break them down into simpler steps.
2. Macs going into Sleep mode cause 40% of all errors - it needs to be turned off using the command ``caffeine -d``.
3. An unstable network is the second biggest cause – use Ethernet on your Mac whenever possible.
4. Always verify after writing operations and use the full file path.
5. Start each Dispatch session with a simple read task to confirm the connection is active.
6. Save important outputs to a file - session memory does not persist after a connection is lost.
7. Dispatch will improve over time - learning it now will give you an early advantage.

1. Question 1:

When Dispatch fails without reporting an error (no error, just unresponsive), what should you check first?

1. A. Check if Claude's server is down - confusing correlation and cause here leads to ineffective strategies.
2. B. Reinstall the Claude application.
3. C. Wait 10 minutes and try again.
4. D. Check if the Dispatch connection is still working by looking at the Dispatch indicator on your phone.

EXPLAIN:

A no-dispatch error usually means the connection was interrupted, but your phone didn't immediately notice. Check your phone's Dispatch indicator – if it shows disconnected, the connection was lost (possibly because your Mac went into Sleep mode or lost its internet connection). Reconnect if necessary. If the indicator shows connected, try a simple check task like 'Are you there?' to verify the connection is working.

2. Question 2:

How can you improve the success rate of complex Dispatch tasks from ~50% to ~80-90%?

1. A. Use a faster internet connection.
2. B. Submitting the same task multiple times until successful – a common mistake leading to suboptimal results.
3. C. Use a shorter prompt.
4. D. Break down complex tasks into simpler steps, focus on a single goal, and verify each step before moving on to the next.

EXPLAIN:

The biggest improvement in reliability comes from breaking it down. A 5-step task that succeeds 50% of the time becomes 5 one-step tasks, each succeeding 80-90% of the time. You also get partial results if any step fails, and you can retry each step individually instead of the entire process.

3. Question 3:

What are the most common causes of Dispatch errors?

1. A. My Mac is out of disk space.
2. B. The Mac goes into Sleep mode, interrupting the connection.
3. C. Claude's AI model is overloaded.
4. D. The phone has lost its mobile signal.

EXPLAIN:

The number one cause of the Dispatch error is your Mac going into Sleep mode. When your Mac sleeps, the Cowork process is paused and the encryption connection is broken. Trying multiple times from your phone won't fix it – your Mac has to wake up. Prevention: Use the command ``caffeinate -d &`` or adjust the Energy settings to prevent your Mac from going into Sleep mode.

Submit your work

Training results

You have completed **0** questions.

-- / --

[Review the lesson](#)

You finished reading the article "**Troubleshooting and best practices in Claude Dispatch**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.