Prof. Philip Koopman

Carnegie Mellon University

Electrical & Computer ENGINEERING

Dependability

Now that's what I call a dead parrot.

– John Cleese (Monty Python)

These tutorials are a simplified introduction, and are not sufficient on their own to achieve system safety. You are responsible for the safety of your system.

Is Your System Dependable?

- Anti-Patterns for Dependability:
  - No concrete dependability goal
  - Confusing reliability vs. availability
  - Mission time is life of product

https://goo.gl/JwwxVH | www.cgpgrey.con

- Can you trust your system?
  - Availability: fraction of up-time
  - Reliability: probability system will complete a mission
  - Other properties, such as:
    - Maintainability
    - Confidentiality
    - Integrity
    - Safety

Availability

- Availability is “up time”

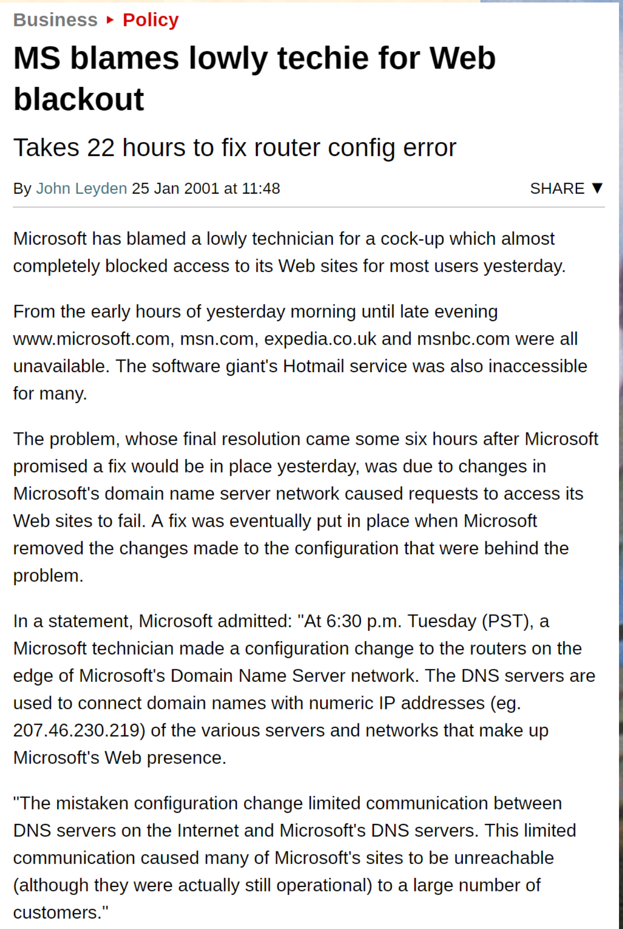
$$Availability = \frac{UpTime}{TotalTime}$$

- Limits to availability
  - Frequency of system failures
    - Redundancy can improve availability
  - Detection & repair time
    - Detect, diagnose, repair failed component, restart the system
    - Time to reconfigure to redundant standby
  - As a practical matter, 99.999% is considered “high availability”
    - 99.999% “Five nines” ➔ ~5 minutes/year down time
    - 99.9999% “Six nines” ➔ 31.5 seconds/year down time

Hours Since
Last System Crash:

0 0 0 0 3

99.9999% Availability Target:
= 2.6 seconds/month downtime



Business ▸ Policy

MS blames lowly techie for Web blackout

Takes 22 hours to fix router config error

By John Leyden 25 Jan 2001 at 11:48 SHARE ▼

Microsoft has blamed a lowly technician for a cock-up which almost completely blocked access to its Web sites for most users yesterday.

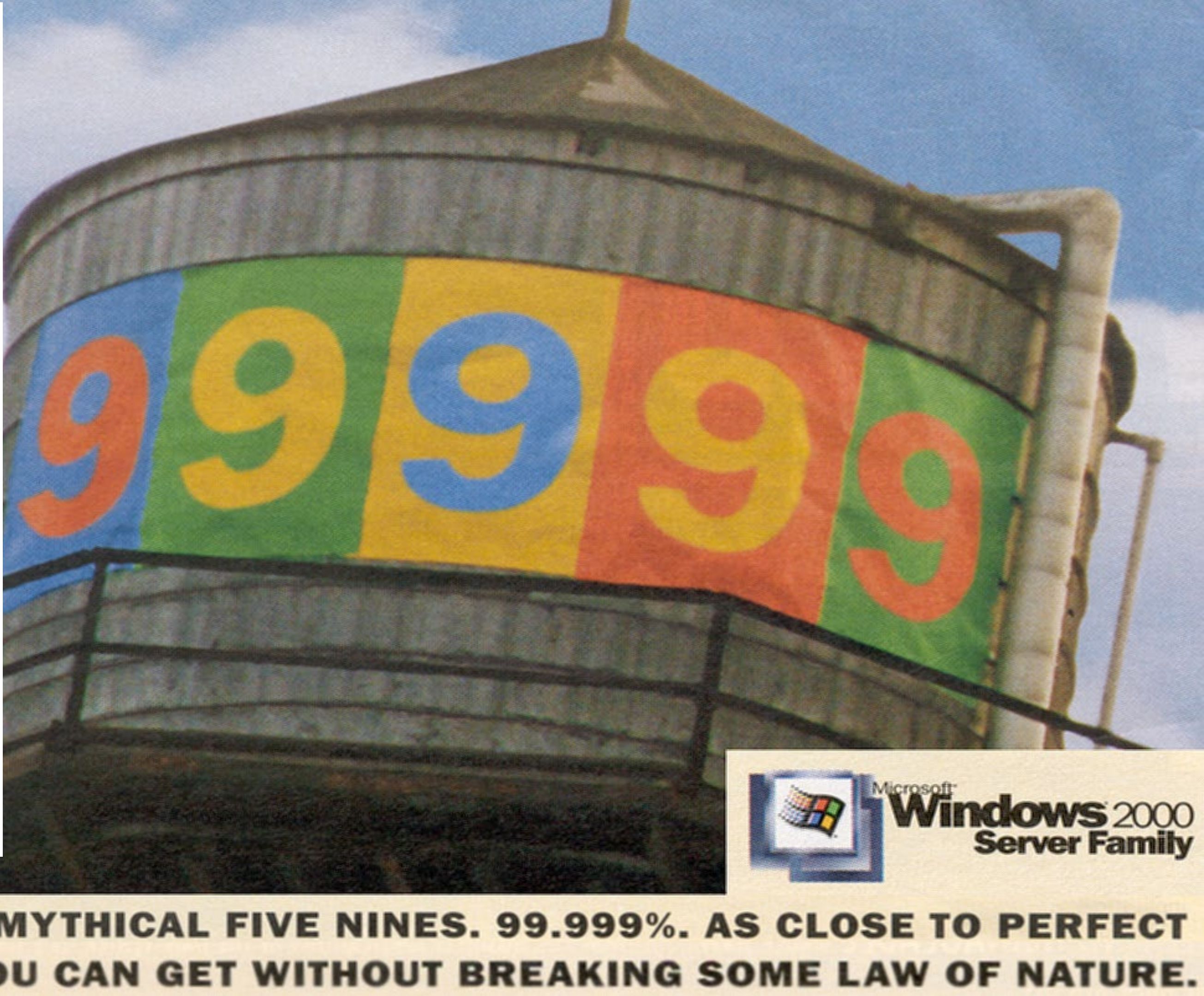
From the early hours of yesterday morning until late evening www.microsoft.com, msn.com, expedia.co.uk and msnbc.com were all unavailable. The software giant's Hotmail service was also inaccessible for many.

The problem, whose final resolution came some six hours after Microsoft promised a fix would be in place yesterday, was due to changes in Microsoft's domain name server network caused requests to access its Web sites to fail. A fix was eventually put in place when Microsoft removed the changes made to the configuration that were behind the problem.

In a statement, Microsoft admitted: "At 6:30 p.m. Tuesday (PST), a Microsoft technician made a configuration change to the routers on the edge of Microsoft's Domain Name Server network. The DNS servers are used to connect domain names with numeric IP addresses (eg. 207.46.230.219) of the various servers and networks that make up Microsoft's Web presence.

"The mistaken configuration change limited communication between DNS servers on the Internet and Microsoft's DNS servers. This limited communication caused many of Microsoft's sites to be unreachable (although they were actually still operational) to a large number of customers."

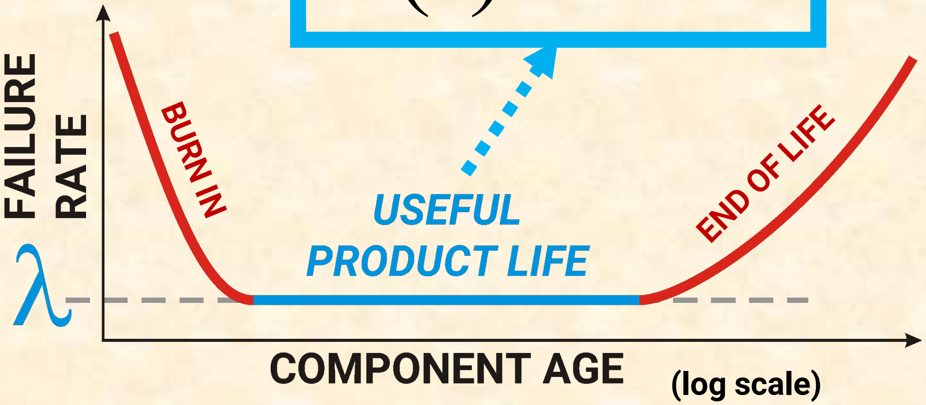
https://www.theregister.co.uk/2001/01/25/ms_blames_lowly_techie/

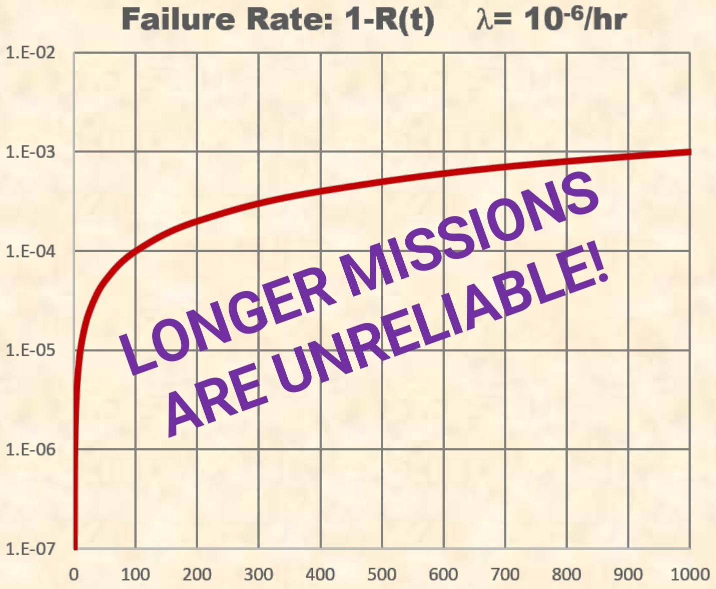

99999
Microsoft Windows 2000 Server Family
THE MYTHICAL FIVE NINES. 99.999%. AS CLOSE TO PERFECT AS YOU CAN GET WITHOUT BREAKING SOME LAW OF NATURE.

Measuring Reliability

- Reliability is based on the concept of a “mission”
  - Reliability R(t): probability system still working since start of mission
  - A mission is t continuous operating hours between diagnostics
  - Constant Failure Rate λ (failures/hr)

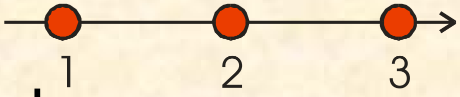
$$\boxed{R(t) = e^{-\lambda t}}$$





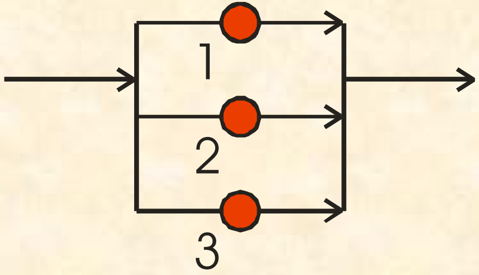
Redundancy Improves Reliability

- Serial reliability
  - Even good components aren't enough
  - E.g.: 0.9 * 0.9 * 0.9 = 0.73



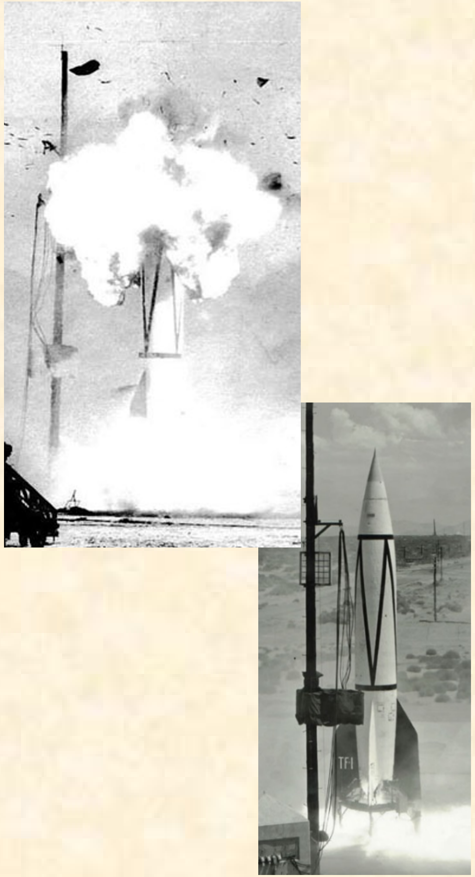
$$R(t)_{SERIAL} = R(t)_1 R(t)_2 R(t)_3 = \prod_i R(t)_i$$

- Parallel reliability
  - Redundancy improves reliability
  - E.g., three @ 0.9 ➔ 0.999



$$R(t)_{PARALLEL} = 1 - \left[(1 - R(t)_1)(1 - R(t)_2)(1 - R(t)_3) \right]$$

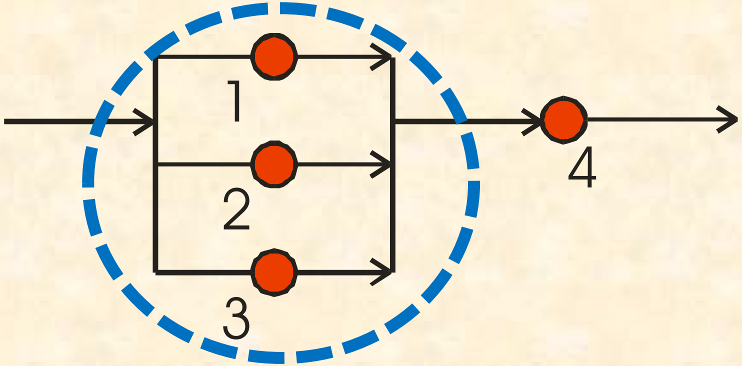
$$R(t)_{PARALLEL} = 1 - \prod_i \left(1 - R(t)_i\right)$$



Example Calculations

Carnegie Mellon University

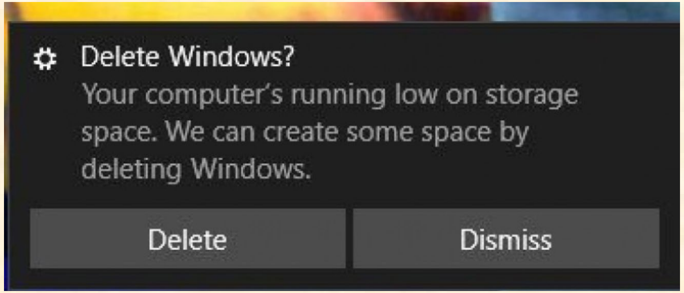
- Reliability at MTBF R(1/lambda) is 36.8%, not 50%. Why?
- What is reliability of this system for 3 hour mission?
  - λ_1 = 7 per million hours
  - λ_2 = 200 per million hours
  - λ_3 = 15000 per million hours
  - λ_4 = 2 per million hours
  - $R(3)_1 = e^{-3*7*10^{-6}} = 0.999979$
  - $R(3)_2 = e^{-3*200*10^{-6}} = 0.999400$
  - $R(3)_3 = e^{-3*15000*10^{-6}} = 0.955997$
  - $R(3)_4 = e^{-3*2*10^{-6}} = \ 0.999994$
  - $R(3)_{PARALLEL} = 1-[(1-R(3)_1)(1-R(3)_2)(1-R(3)_3)] = 0.999\,999\,999\,45$
  - $R(3)_{TOTAL} = R(3)_{PARALLEL}\ R(3)_4 = 0.999\,999\,999\,45 * 0.999994 = 0.999994$



$R(3)_{PARALLEL}$

https://bit.ly/2pzdJ7p

https://bit.ly/33EwQf0



Delete Windows?

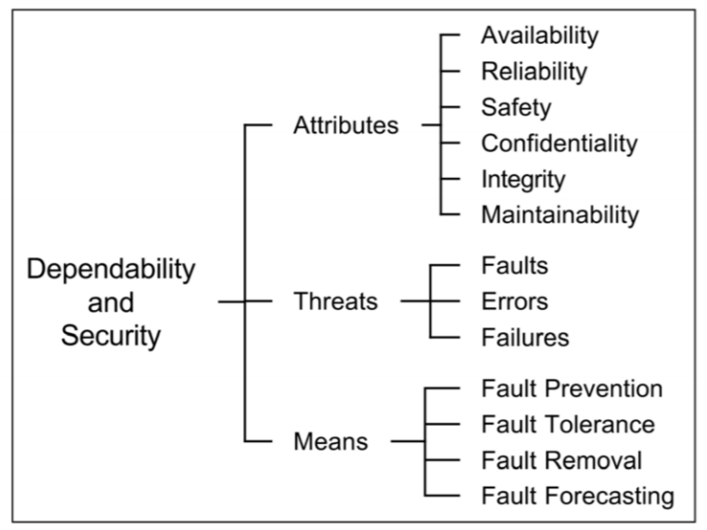
Your computer's running low on storage space. We can create some space by deleting Windows.

https://bit.ly/2OXaH7I

https://goo.gl/ww85xe

Other Aspects of Dependability

- Availability: up-time fraction
- Reliability: no failures
- Safety: no mishaps, no loss events
- Confidentiality: no disclosures
- Integrity: no corruption of state
- Maintainability: system can be fixed
  - E.g., “80% of failures can be fixed in 1 hour”



https://goo.gl/SyV4uZ

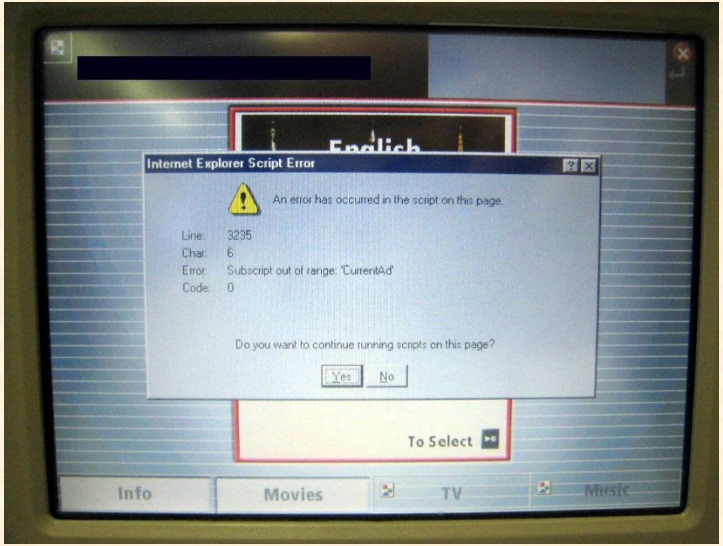
- Fault progression:
  - A fault is something that goes wrong (e.g., bit flip)
  - An error is an activated fault (e.g., flipped bit is read and used in a calculation)
  - A failure is when system does not provide required service (e.g., incorrect output)

A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE Trans. Dependability, Jan-Mar 2004, pp. 11-33

Best Practices For Dependability

- Specify a dependability target
  - "Never fails" is unrealistic
  - Do you care about reliability or availability?

- Minimize impact of any faults
  - Fault ➔ Error ➔ System Failure
  - Parallel redundancy usually helps
  - Fast detection and reconfiguration



[redacted]

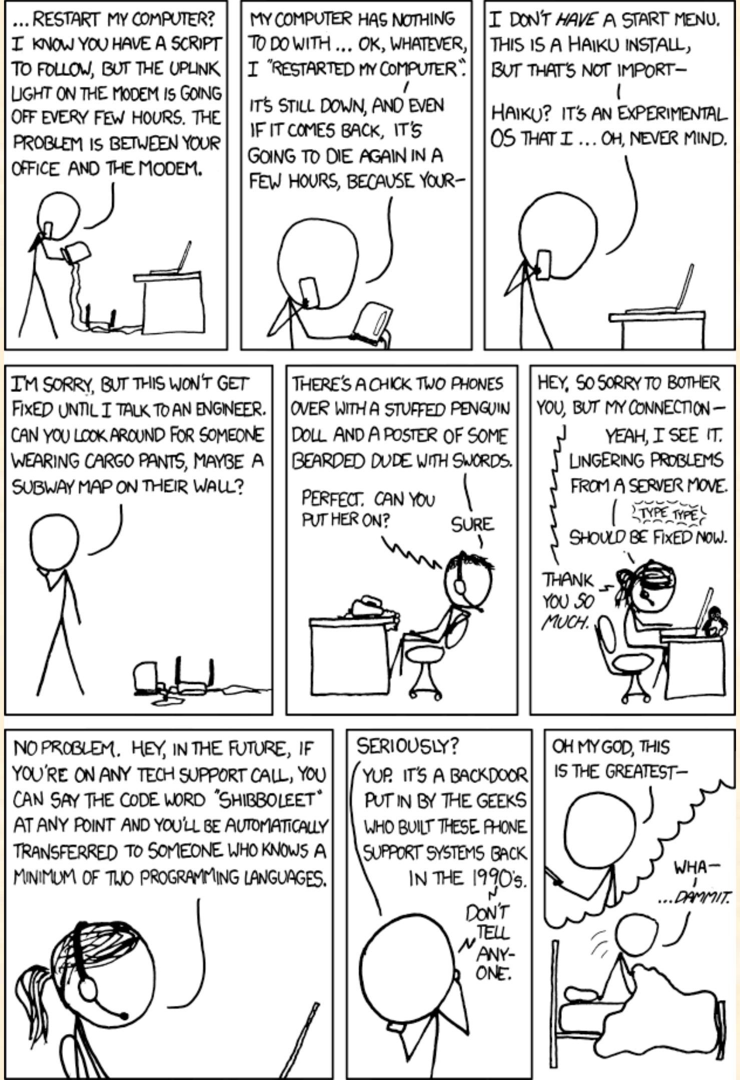
```
Internet Explorer Script Error

An error has occurred in the script on this page.

Line:   3235
Char:   6
Error:  Subscript out of range: 'CurrentAd'
Code:   0

Do you want to continue running scripts on this page?
```

- Pitfalls:
  - Long missions without redundancy diagnosis/repair
  - Non-redundant components are weak spot ➔ single points of failure
    - Software failures are generally neither random nor independent
  - Security matters too: attacks; outages for patches


... RESTART MY COMPUTER? I KNOW YOU HAVE A SCRIPT TO FOLLOW, BUT THE UPLINK LIGHT ON THE MODEM IS GOING OFF EVERY FEW HOURS. THE PROBLEM IS BETWEEN YOUR OFFICE AND THE MODEM.
MY COMPUTER HAS NOTHING TO DO WITH ... OK, WHATEVER, I "RESTARTED MY COMPUTER".
IT'S STILL DOWN, AND EVEN IF IT COMES BACK, IT'S GOING TO DIE AGAIN IN A FEW HOURS, BECAUSE YOUR—
I DON'T *HAVE* A START MENU. THIS IS A HAIKU INSTALL, BUT THAT'S NOT IMPORT—
HAIKU? IT'S AN EXPERIMENTAL OS THAT I ... OH, NEVER MIND.
I'M SORRY, BUT THIS WON'T GET FIXED UNTIL I TALK TO AN ENGINEER. CAN YOU LOOK AROUND FOR SOMEONE WEARING CARGO PANTS, MAYBE A SUBWAY MAP ON THEIR WALL?
THERE'S A CHICK TWO PHONES OVER WITH A STUFFED PENGUIN DOLL AND A POSTER OF SOME BEARDED DUDE WITH SWORDS.
PERFECT. CAN YOU PUT HER ON?
SURE
HEY, SO SORRY TO BOTHER YOU, BUT MY CONNECTION—
YEAH, I SEE IT. LINGERING PROBLEMS FROM A SERVER MOVE.
TYPE TYPE
SHOULD BE FIXED NOW.
THANK YOU SO MUCH.
NO PROBLEM. HEY, IN THE FUTURE, IF YOU'RE ON ANY TECH SUPPORT CALL, YOU CAN SAY THE CODE WORD "SHIBBOLEET" AT ANY POINT AND YOU'LL BE AUTOMATICALLY TRANSFERRED TO SOMEONE WHO KNOWS A MINIMUM OF TWO PROGRAMMING LANGUAGES.
SERIOUSLY?
YUP. IT'S A BACKDOOR PUT IN BY THE GEEKS WHO BUILT THESE PHONE SUPPORT SYSTEMS BACK IN THE 1990's.
DON'T TELL ANYONE.
OH MY GOD, THIS IS THE GREATEST—
WHA—
...DAMMIT.

https://xkcd.com/806/