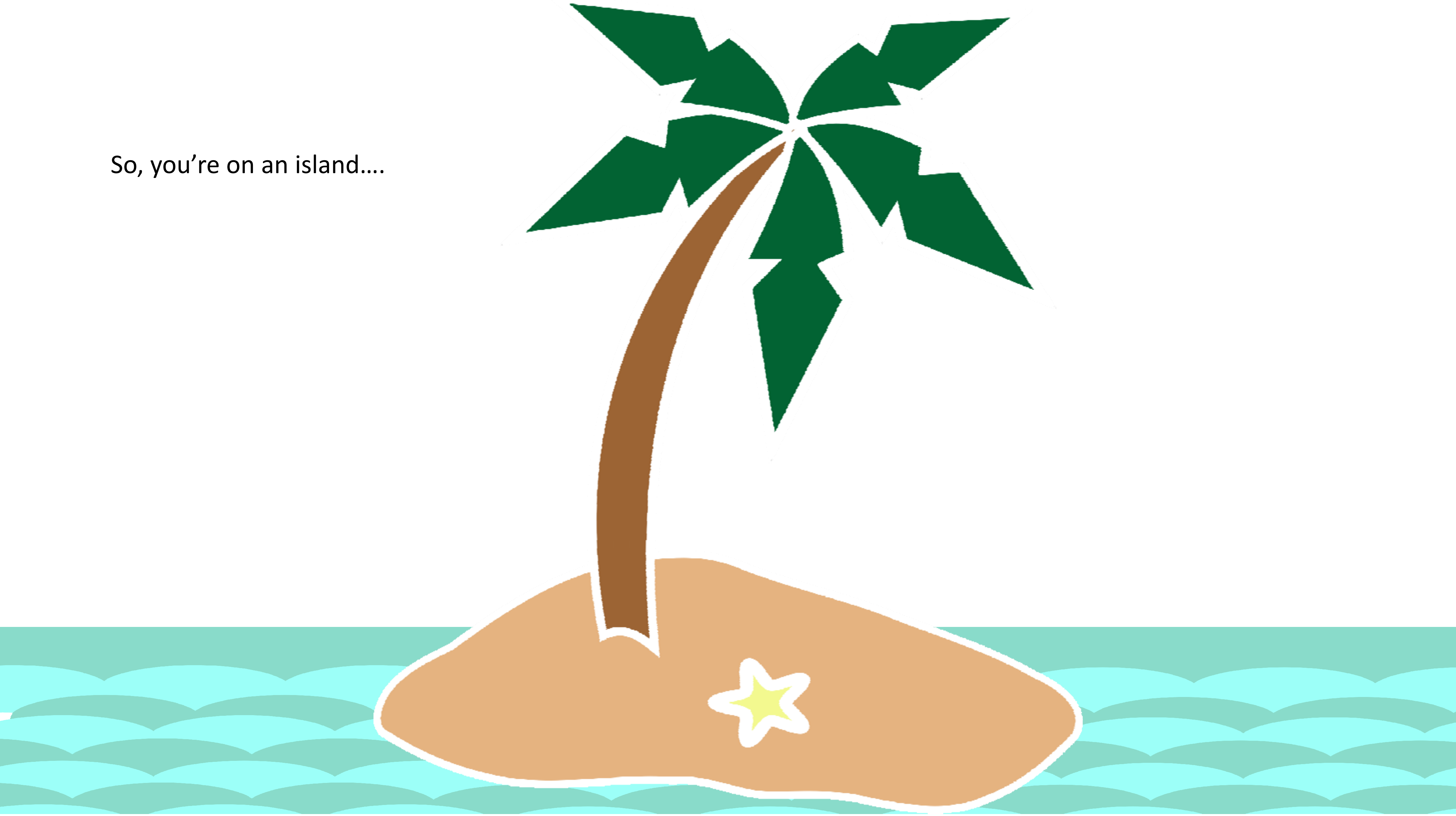
Making the best of satellite links to Pacific Islands

Ulrich Speidel, Lei Qian, ‘Etuate Cocker, Péter Vingelmann, Janus Heide, Muriel Médard

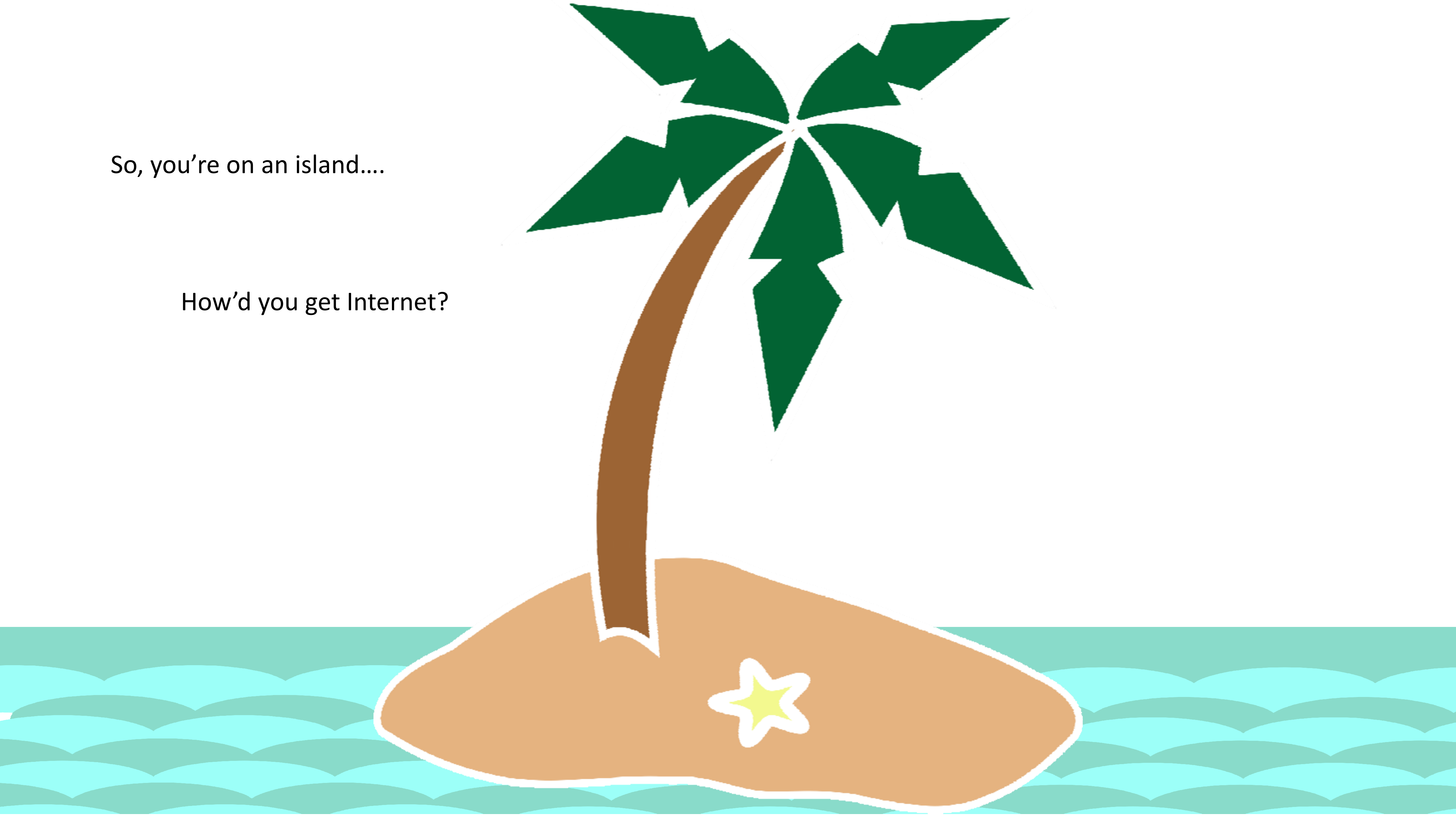
ulrich@cs.auckland.ac.nz

So, you're on an island….



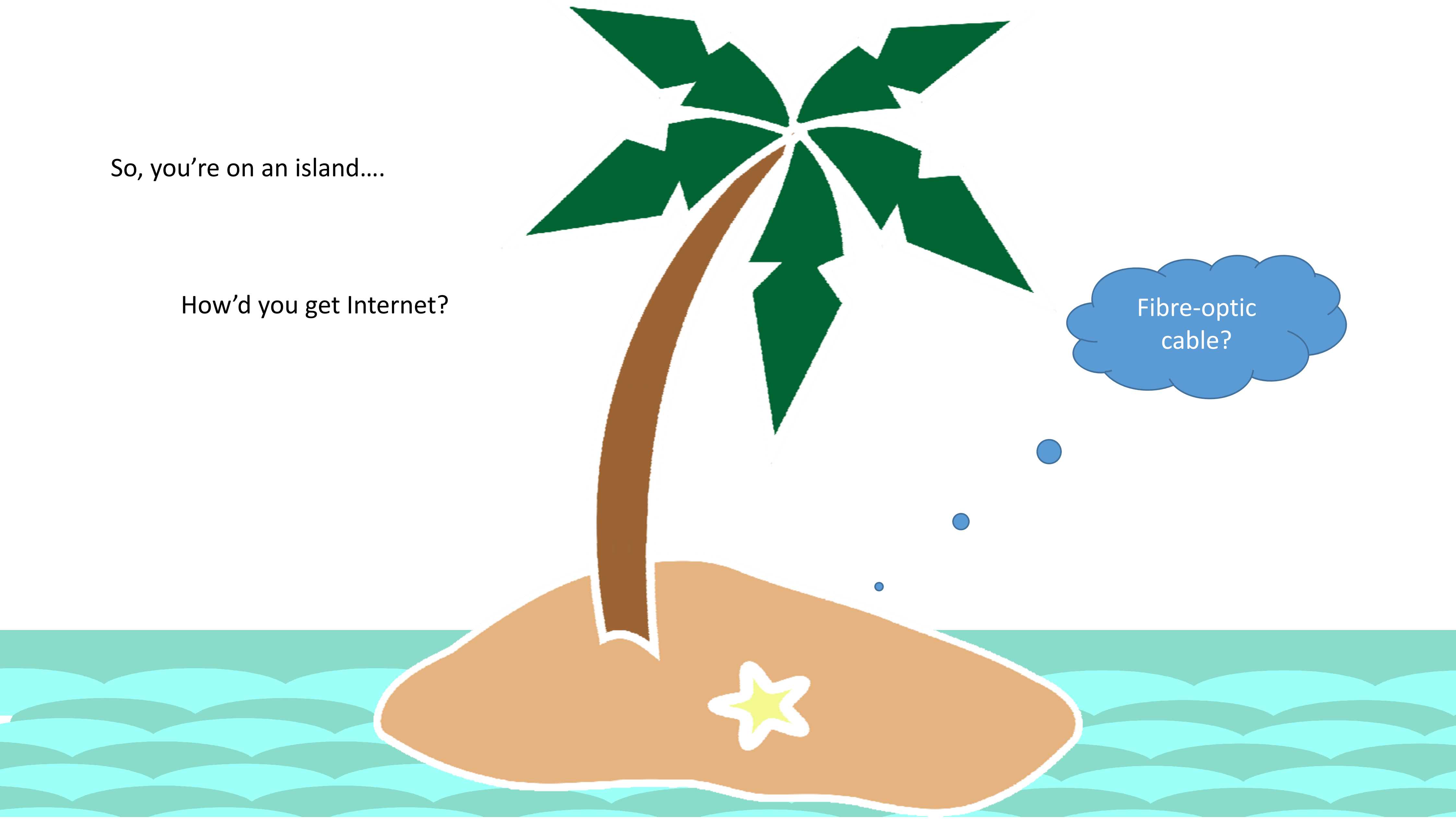
So, you're on an island….

How'd you get Internet?



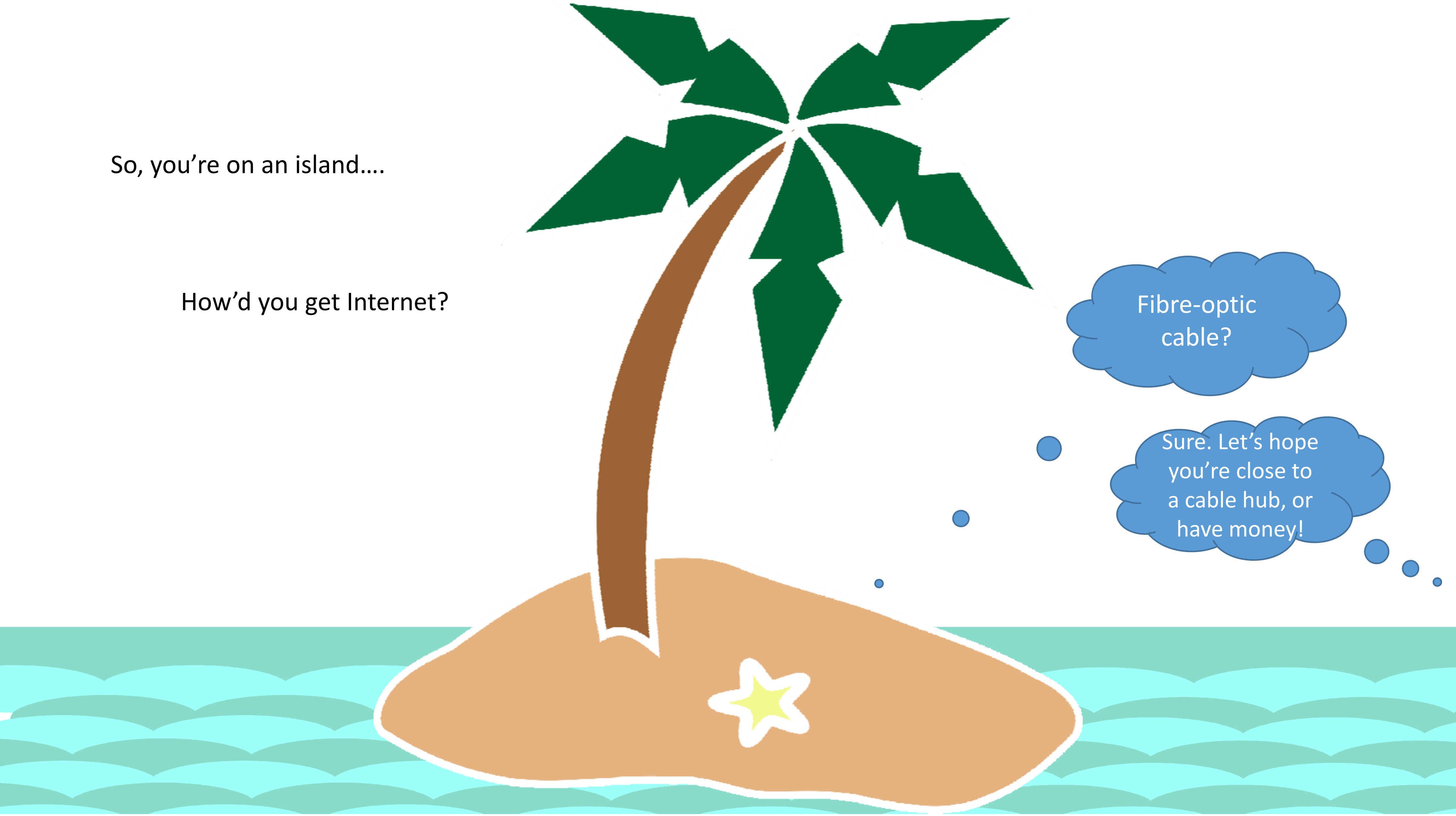
So, you're on an island….

How'd you get Internet?



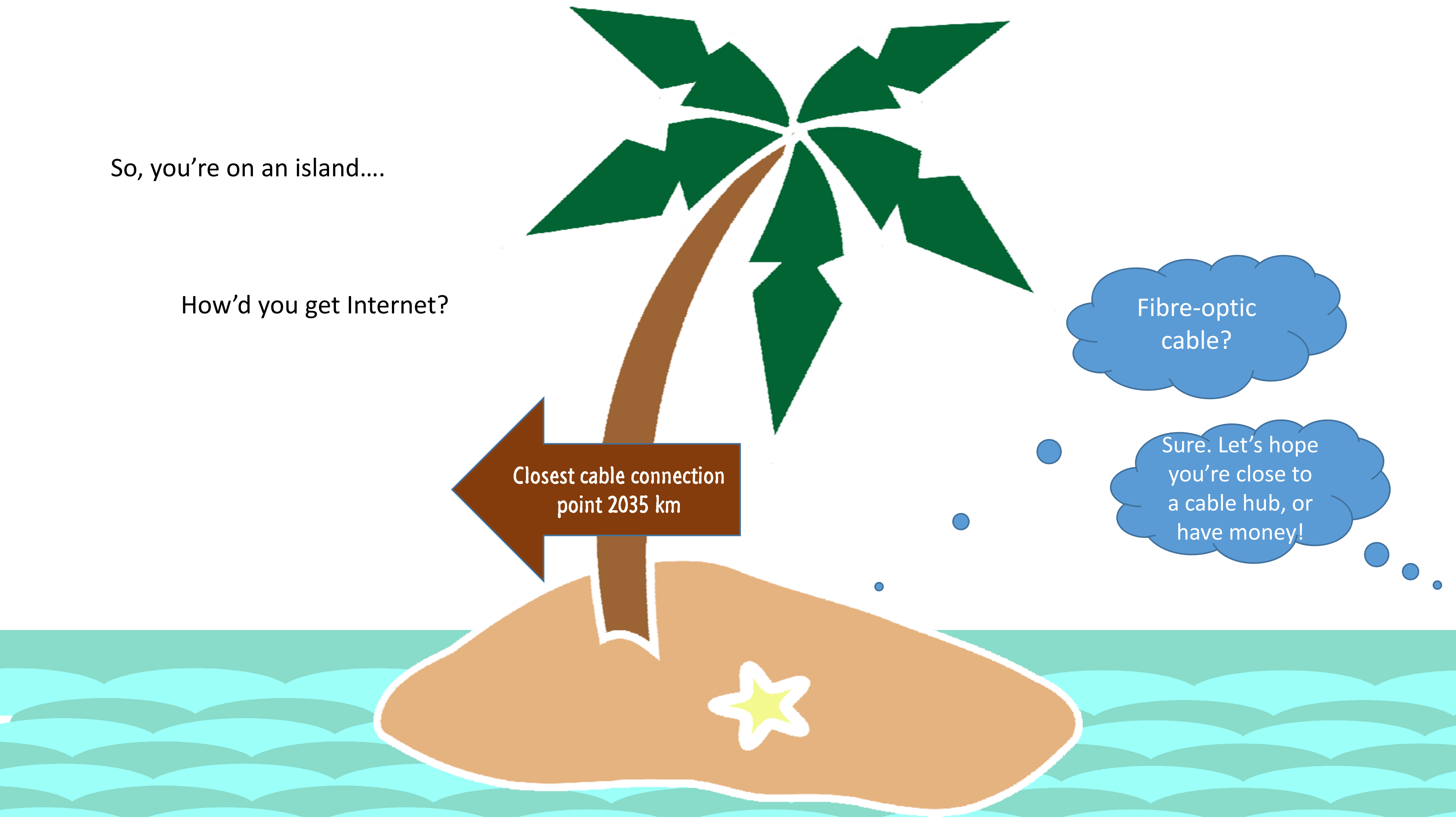
So, you’re on an island….

How’d you get Internet?



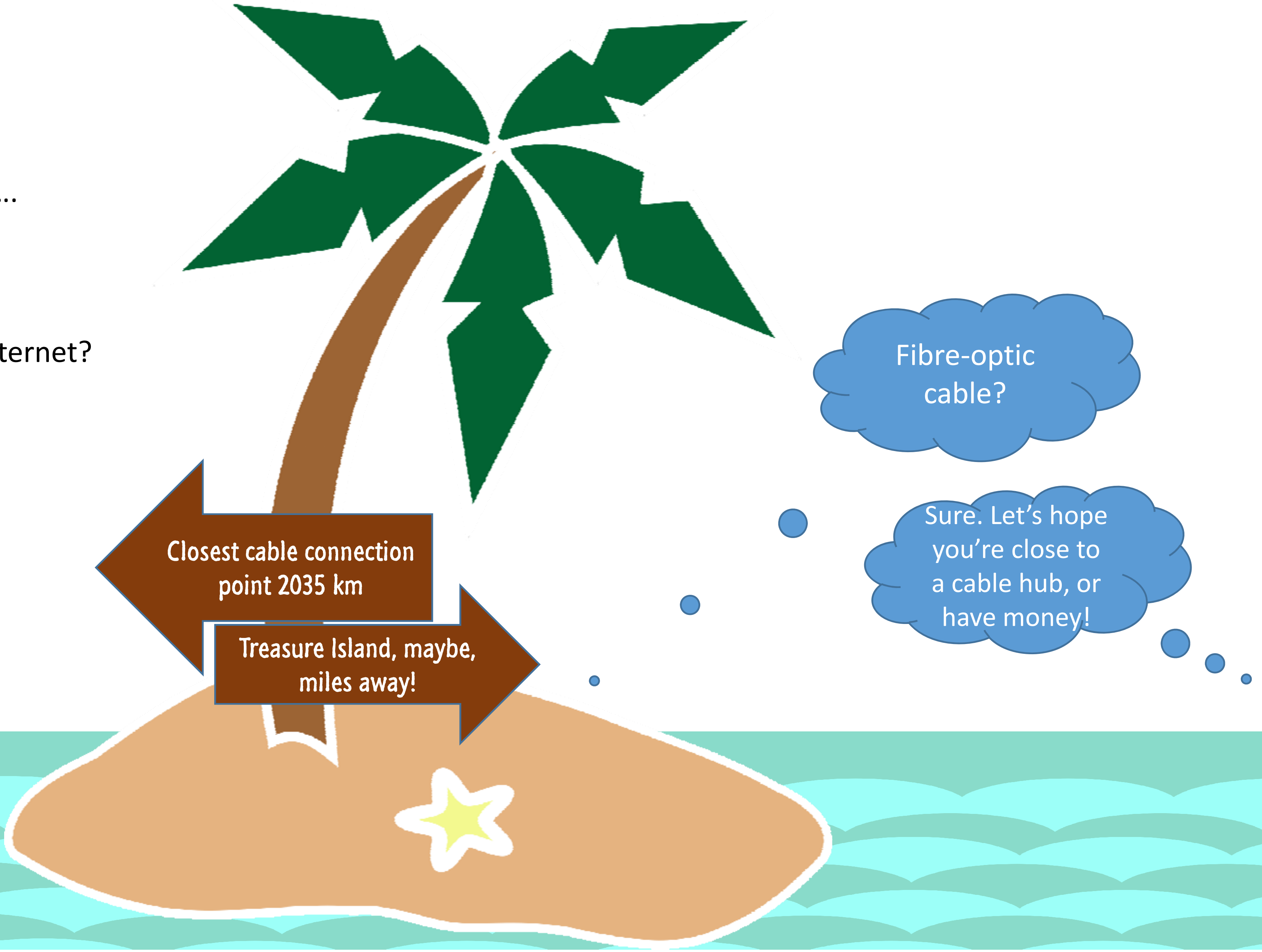
So, you’re on an island….

How’d you get Internet?



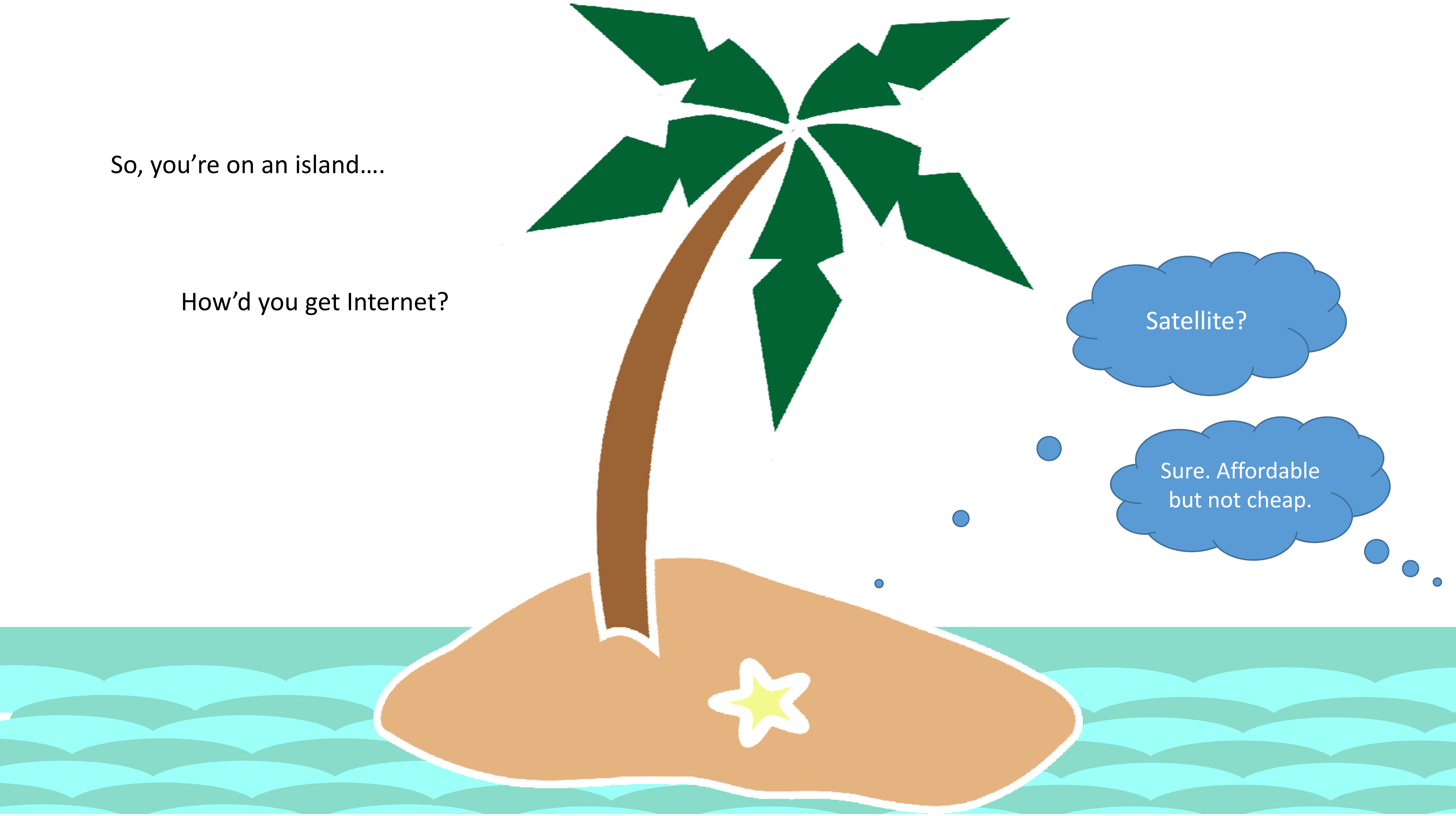
So, you’re on an island….

How’d you get Internet?



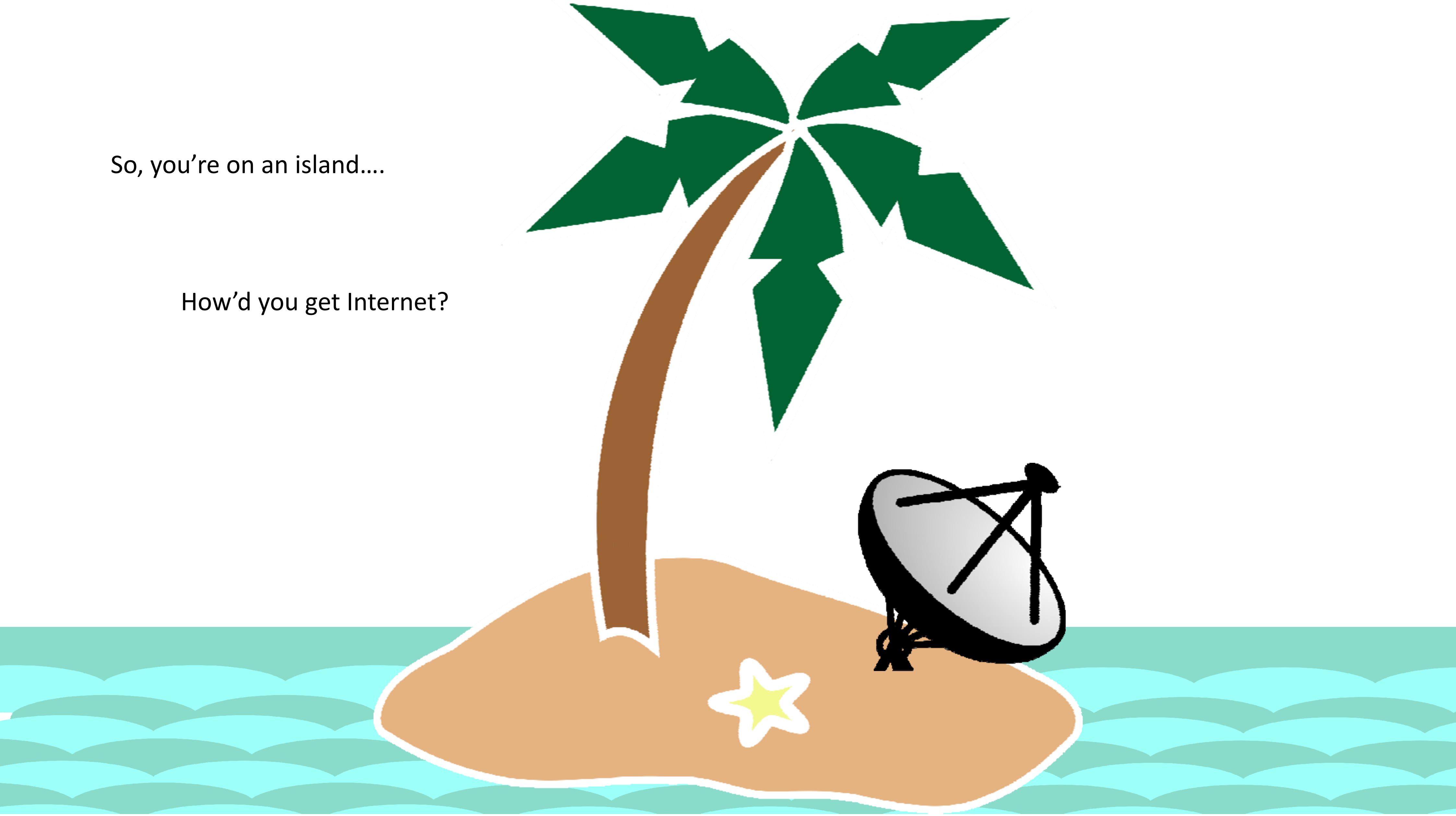
So, you’re on an island….

How’d you get Internet?



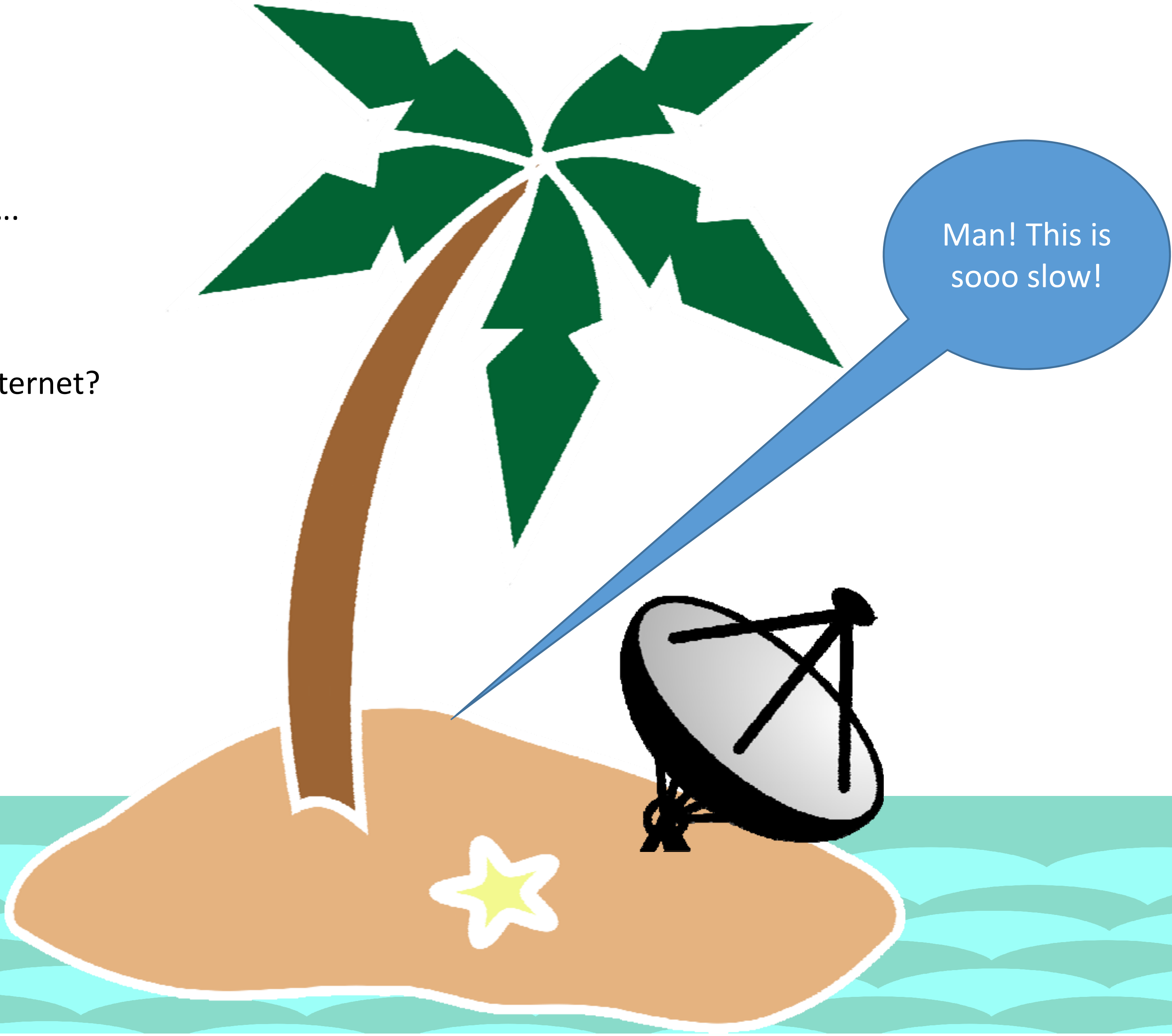
So, you're on an island….

How'd you get Internet?



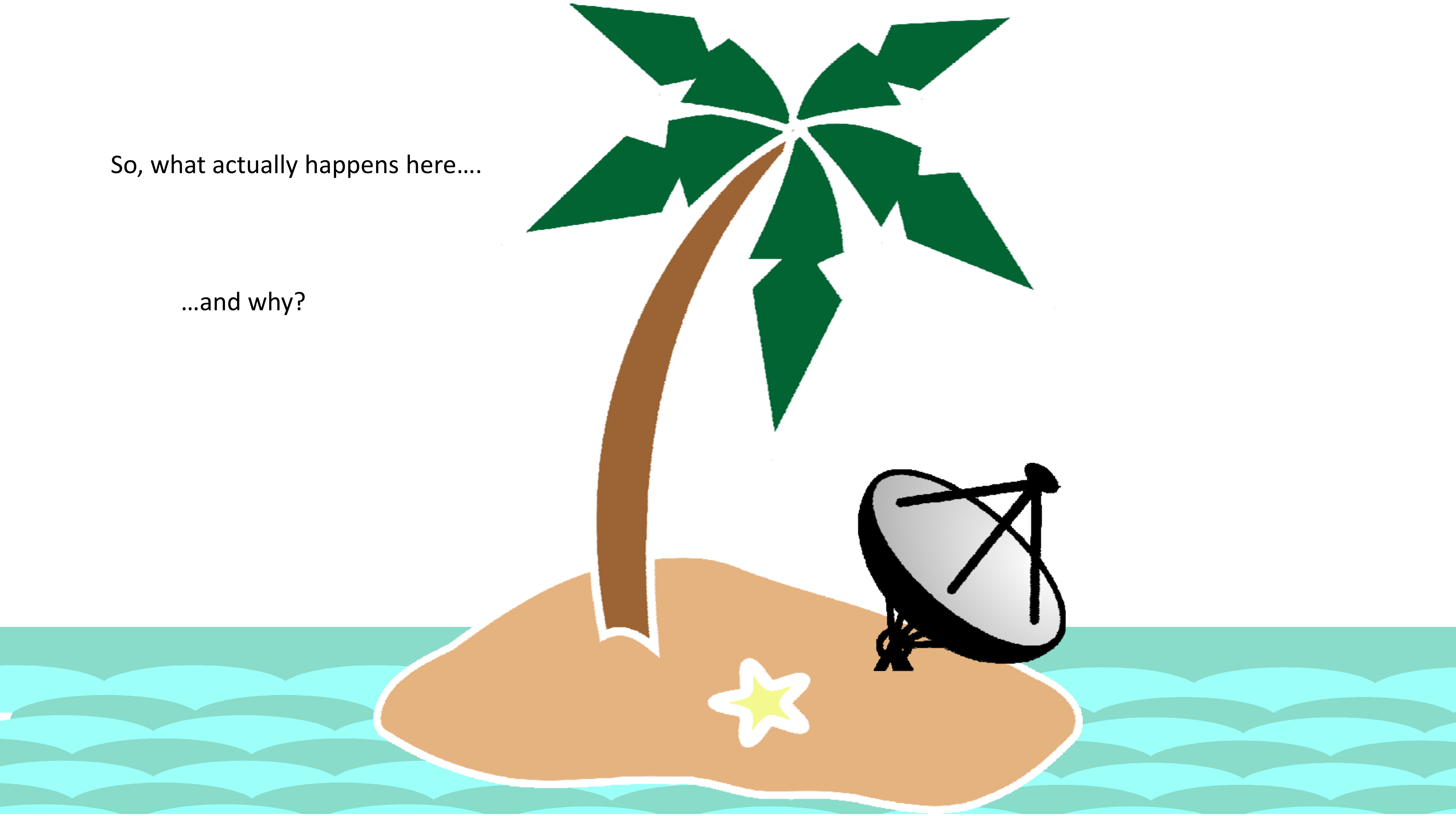
So, you’re on an island….

How’d you get Internet?



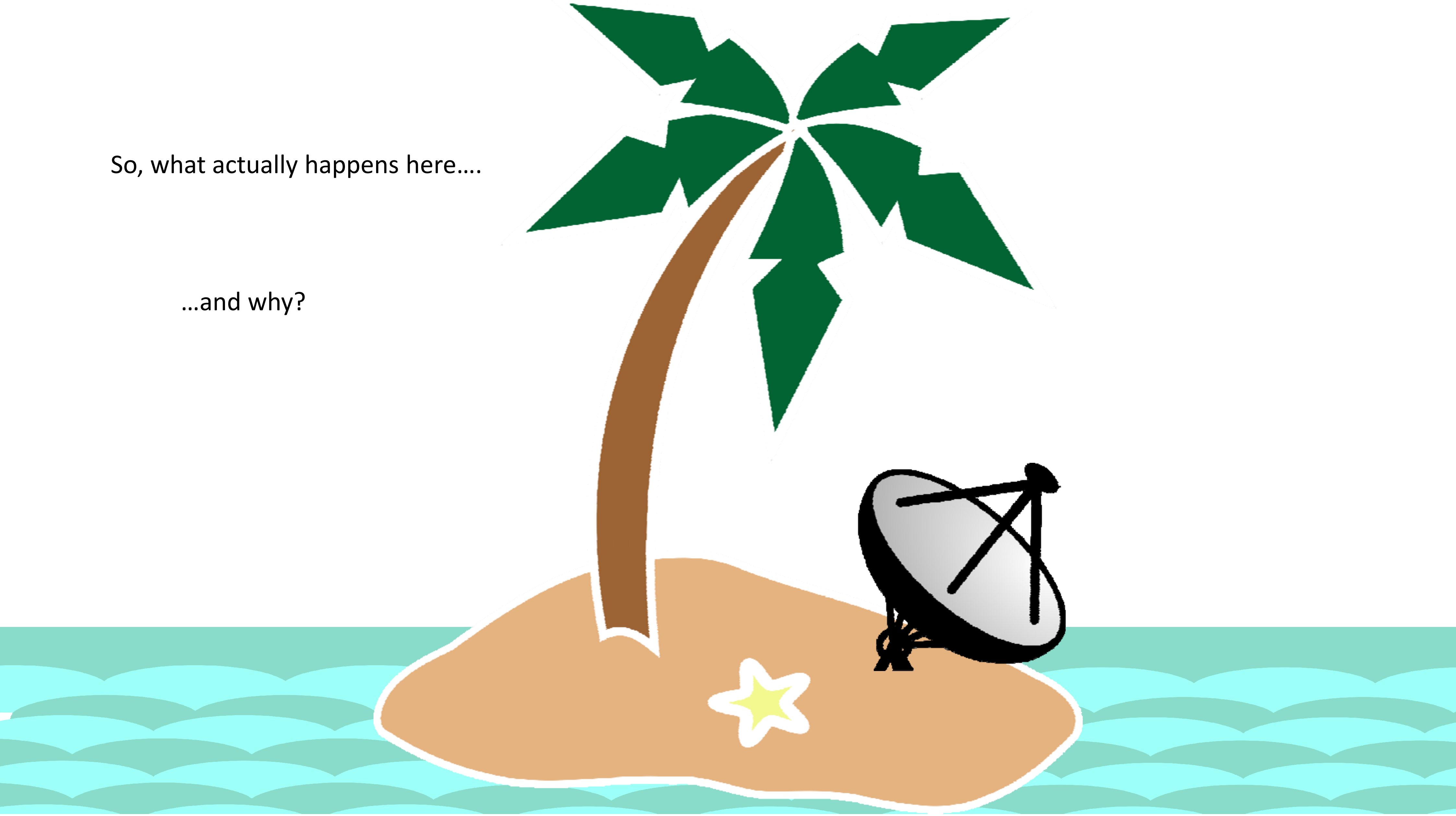
So, what actually happens here….

…and why?



So, what actually happens here….

…and why?



Satellite links are not born equal

Geostationary (~500 ms one-way latency)…

Satellite links are not born equal

Geostationary (~500 ms one-way latency)…

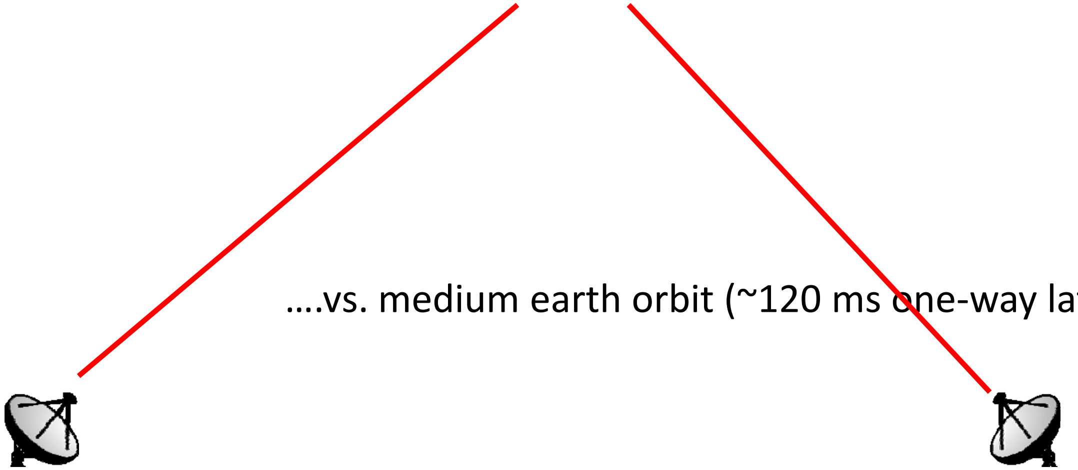
Satellite links are not born equal

….vs. medium earth orbit (~120 ms one-way latency)

Geostationary (~500 ms one-way latency)…

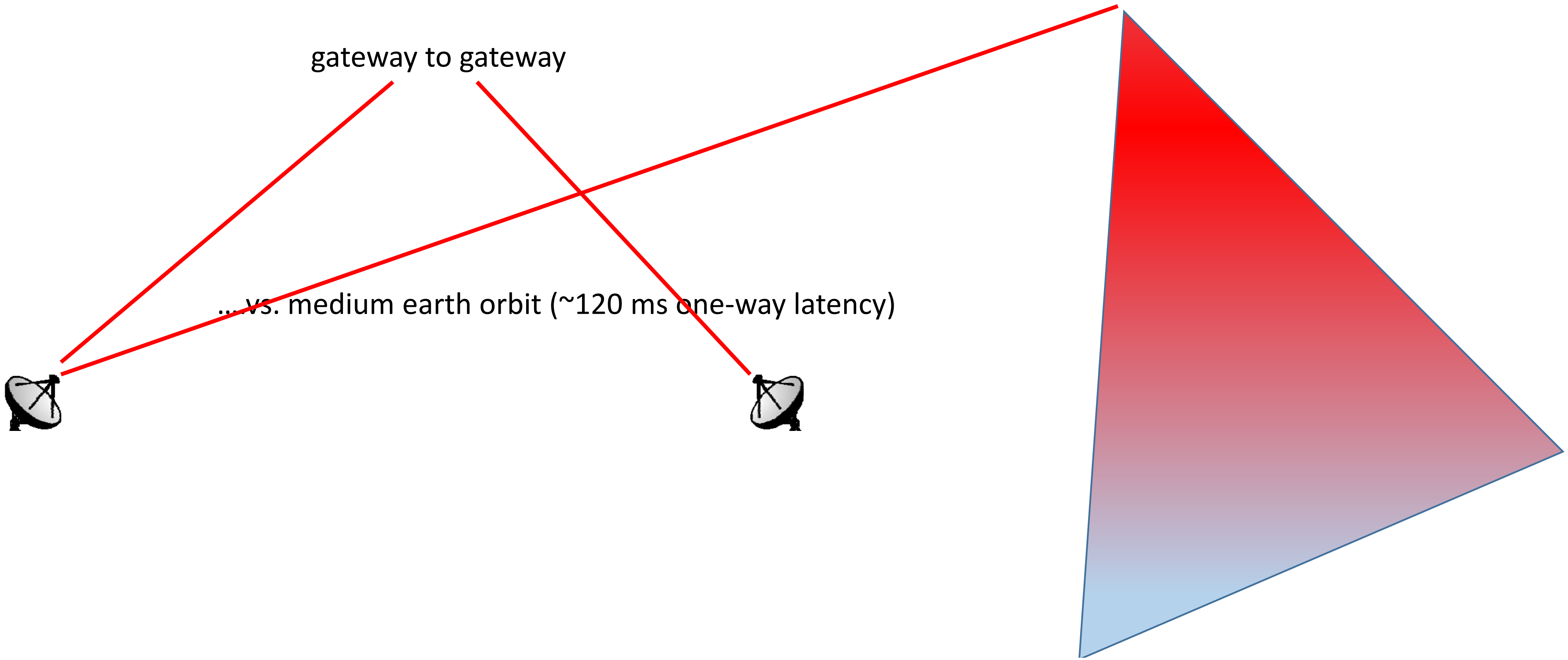
Satellite links are not born equal

gateway to gateway



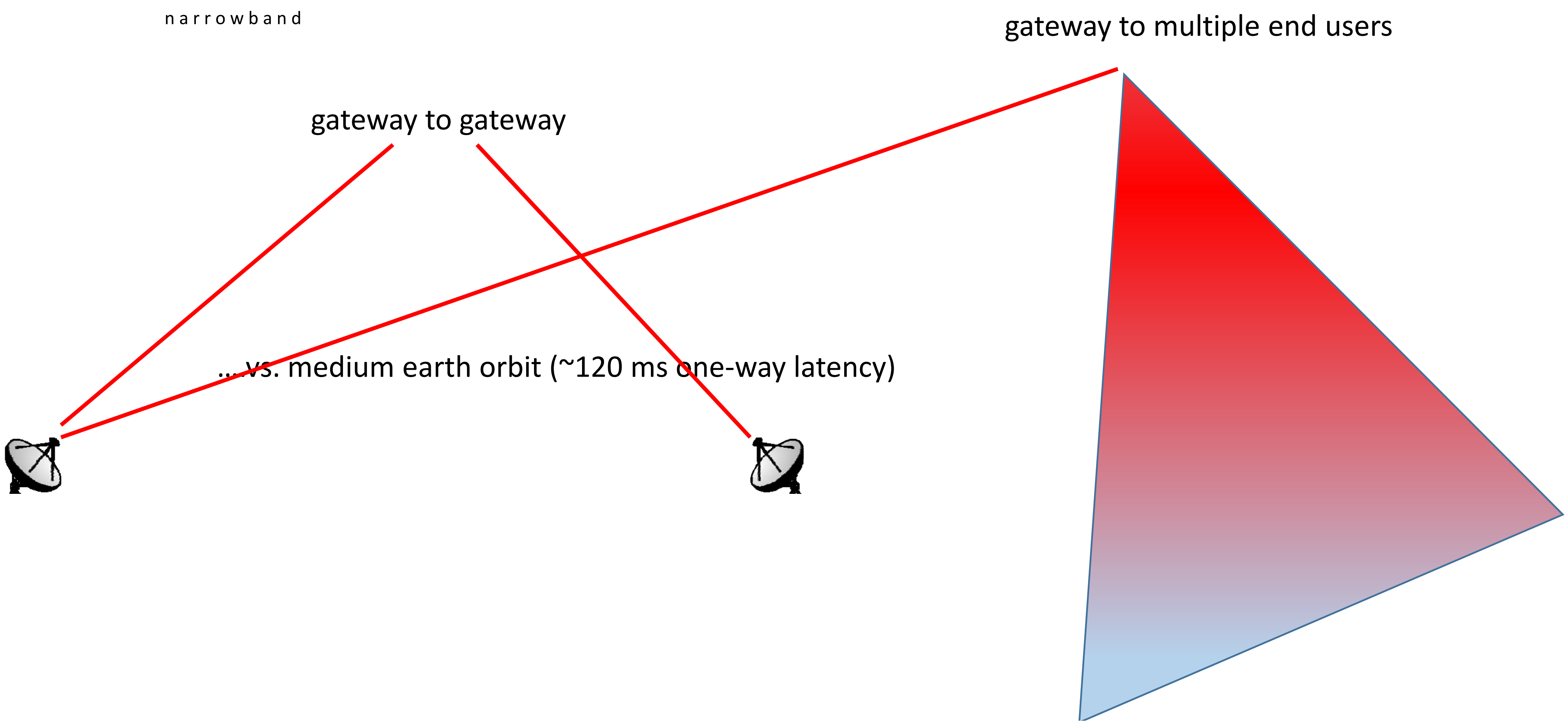
….vs. medium earth orbit (~120 ms one-way latency)

Satellite links are not born equal



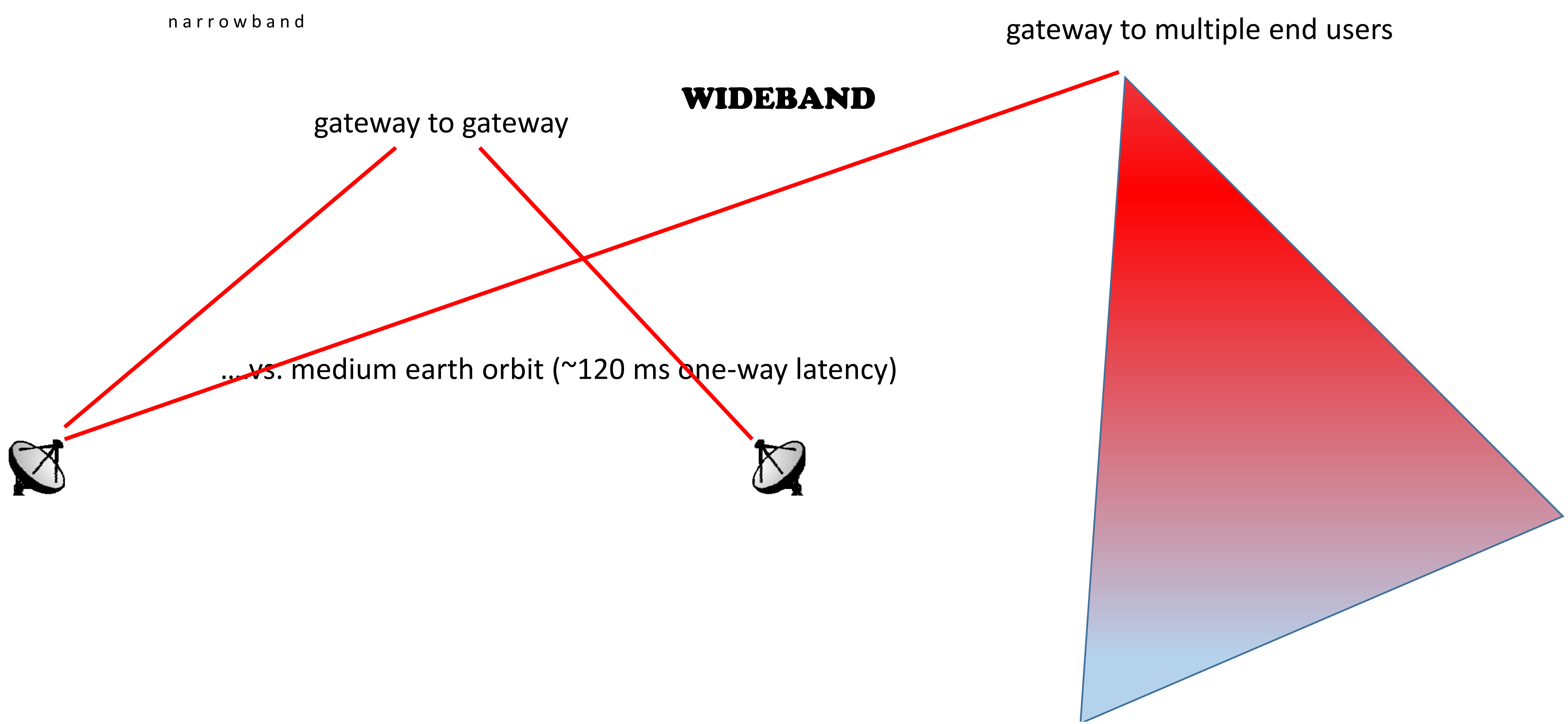
Geostationary (~500 ms one-way latency)…

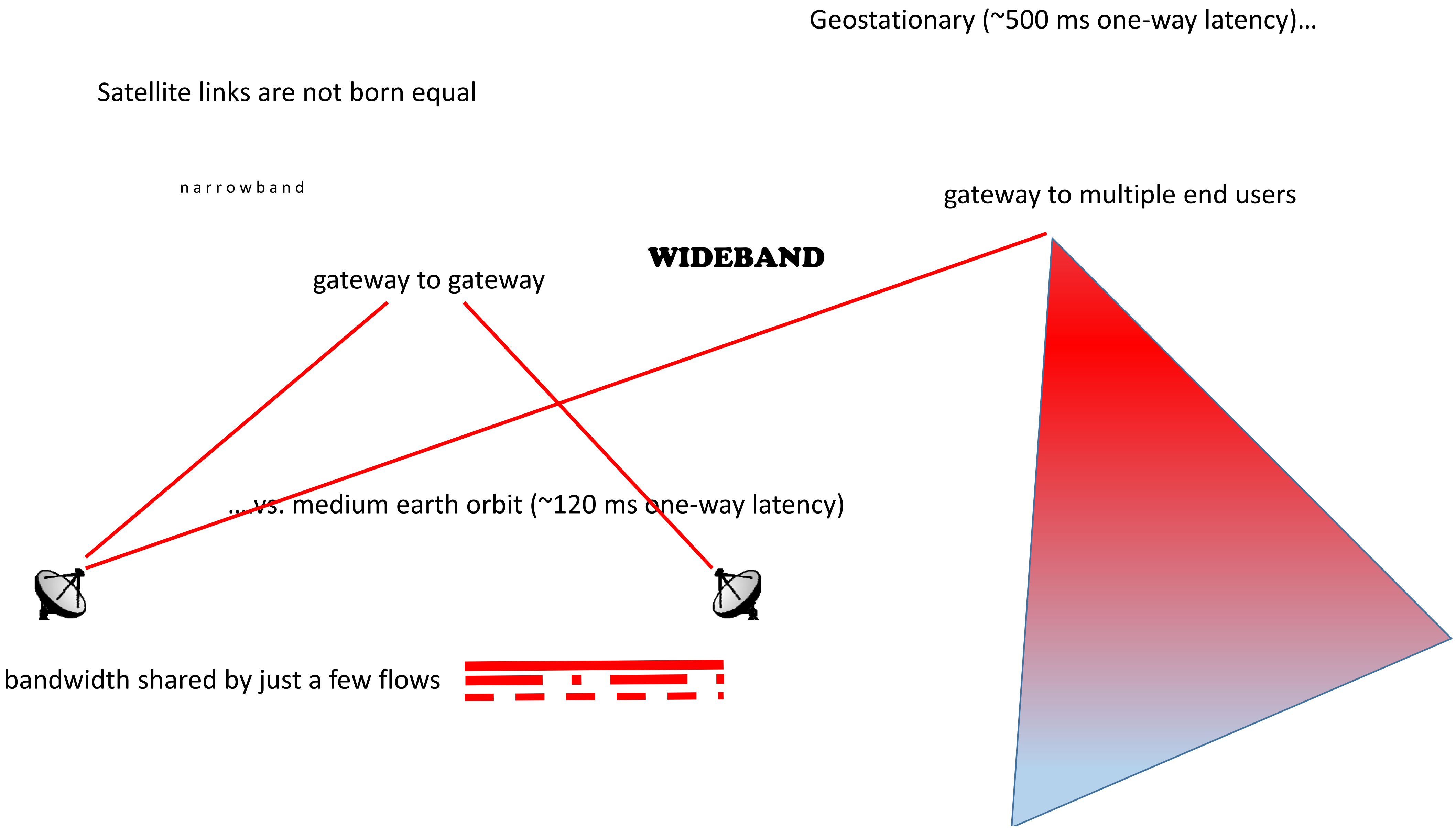
Satellite links are not born equal



Geostationary (~500 ms one-way latency)…

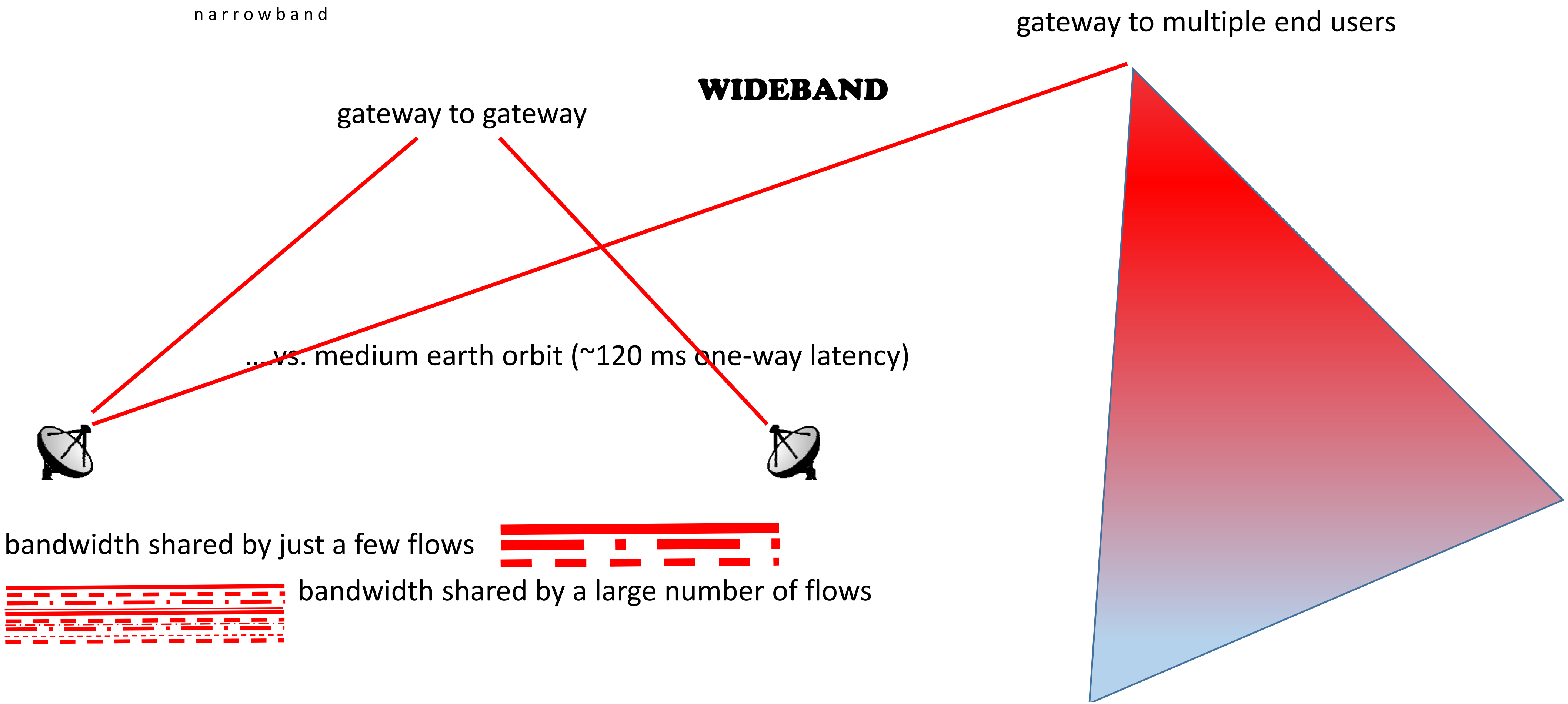
Satellite links are not born equal




Geostationary (~500 ms one-way latency)…
Satellite links are not born equal
narrowband
gateway to multiple end users
WIDEBAND
gateway to gateway
…vs. medium earth orbit (~120 ms one-way latency)
bandwidth shared by just a few flows

Geostationary (~500 ms one-way latency)…

Satellite links are not born equal



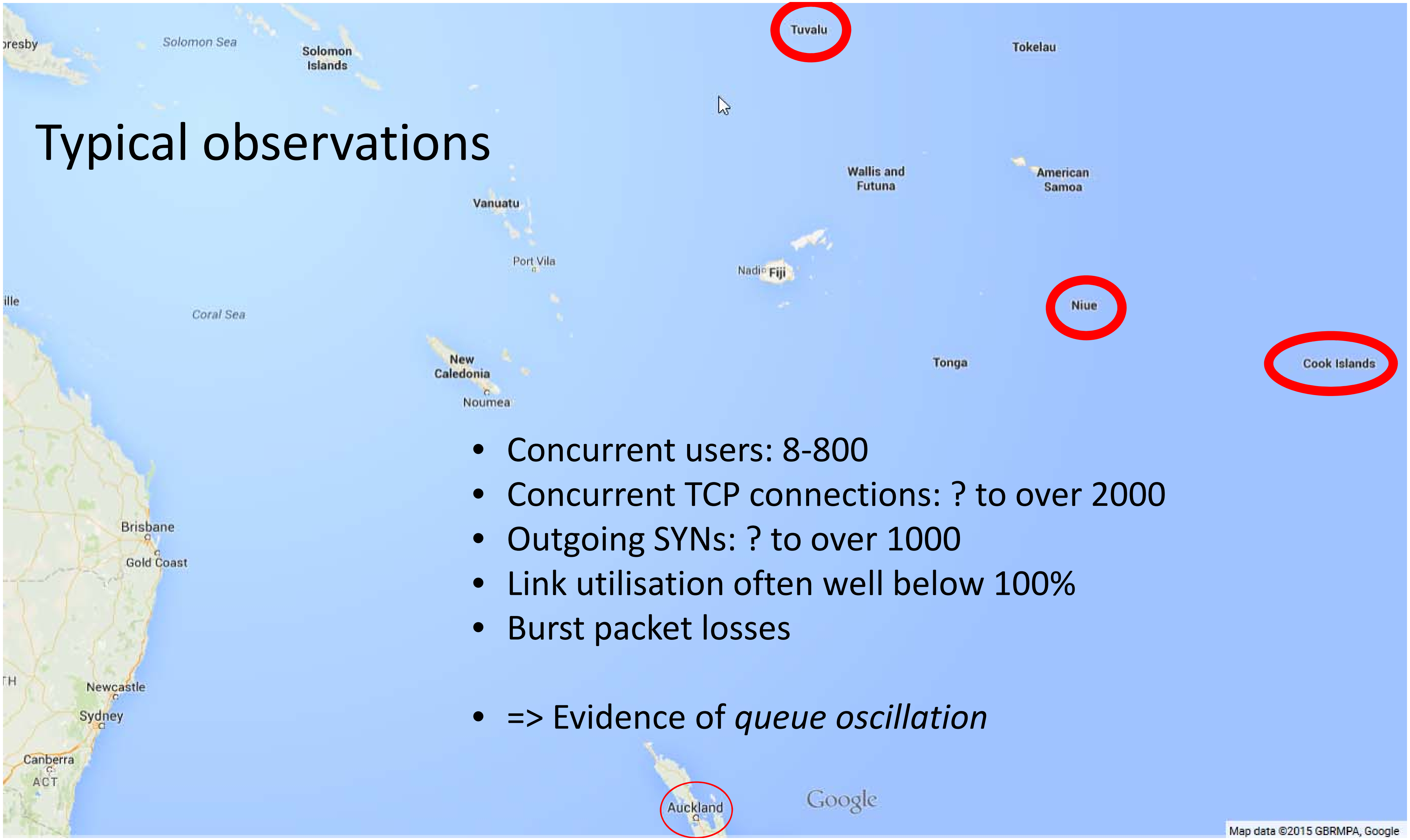
Our partners in the Pacific

- Bluesky Cook Islands
- Internet Niue
- Tuvalu Telecom
- PICISOC

Typical setups

- GEO link from off-island gateway to gateway on island
- MEO link to Rarotonga and Aitutaki in the Cook Islands
- “Pipe model”: All IP traffic to/from the island shares the same channel
- Provisioned bandwidths (inbound traffic): 8-332 Mbps

Typical observations

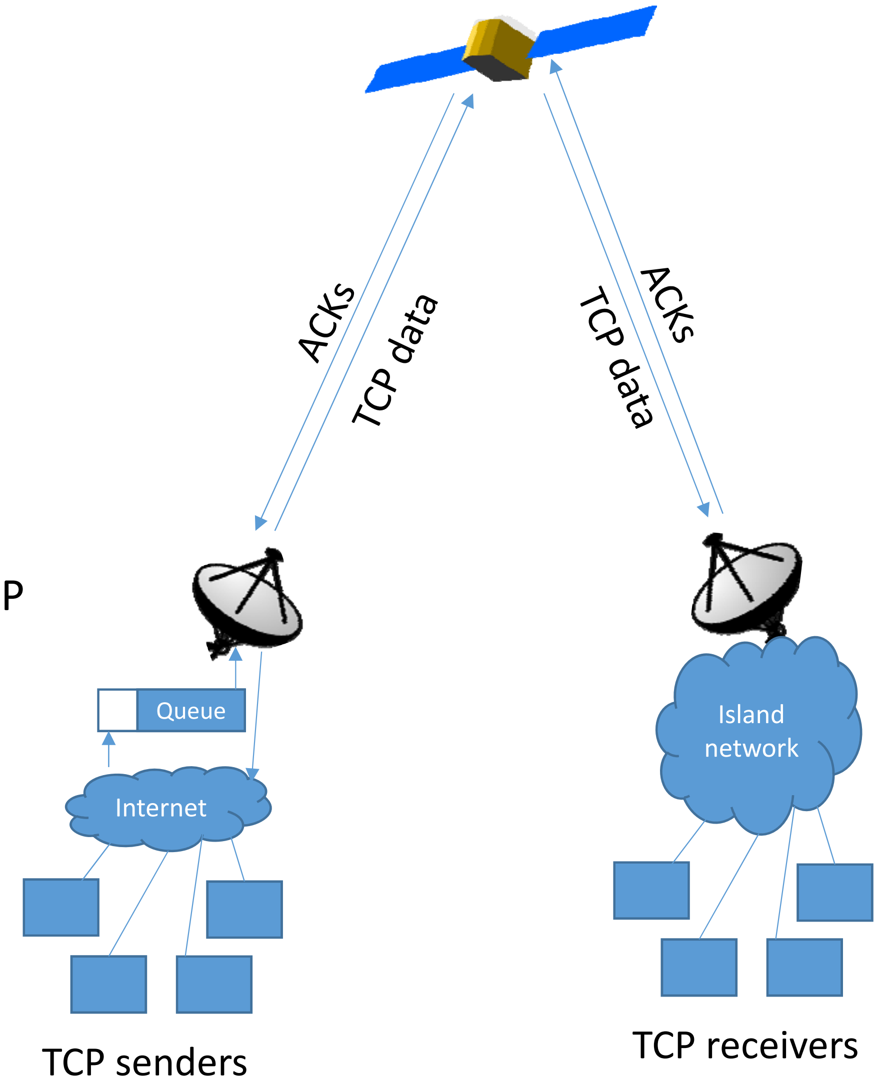


- Concurrent users: 8-800
- Concurrent TCP connections: ? to over 2000
- Outgoing SYNs: ? to over 1000
- Link utilisation often well below 100%
- Burst packet losses

- => Evidence of *queue oscillation*

TCP queue oscillation

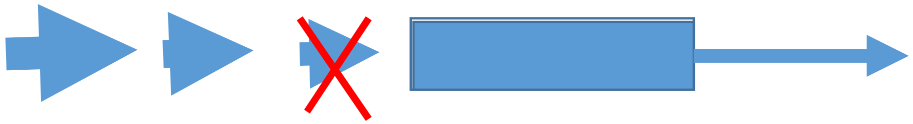
- *Multiple* TCP senders remotely send traffic to the sat gate
- Sat link is a *bottleneck*. *Queue* at sat gate acts like a *funnel*.
- TCP sender cannot see queue state directly
- Feedback on queue state goes via the satellite to remote TCP receivers, and from there back to the senders
- Long delays: >500 ms on GEO, >125 ms on MEO
- Queue can **oscillate** between empty and overflow
- Complicating factors: TCP slow start, exponential back-off



The four phases of queue oscillation

1. Sat gate queue not full. TCP senders receive ACKs, increase congestion window. Queue builds up.

2. Sat gate queue full. New packets arriving are dropped. Senders still receive ACKs and send more data in the direction of the queue. Queue continues to overflow: *burst losses*



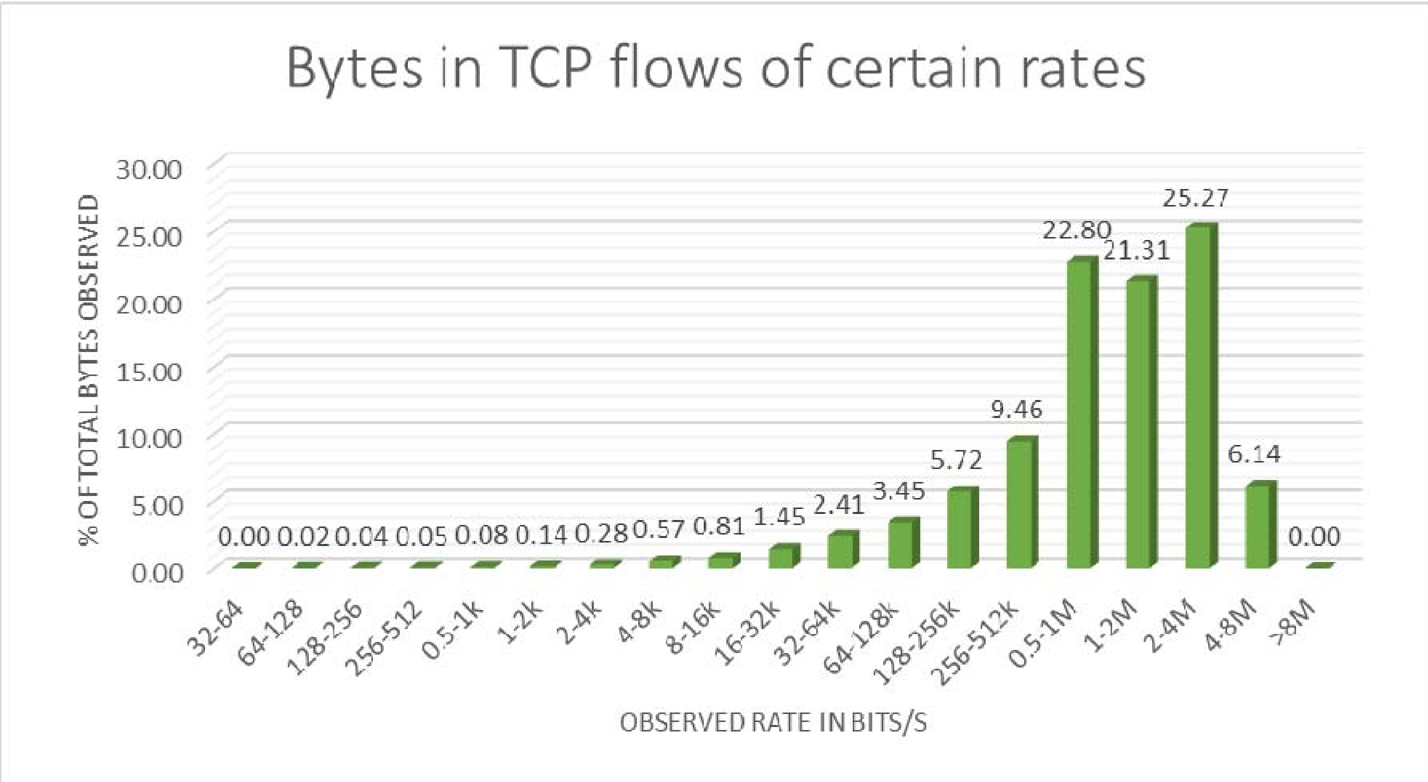
3. ACKs from dropped packets become overdue. Senders throttle back. Packet arrival at queue slows to a trickle. Queue drains.

4. Queue clears completely. Link sits idle for part of the time, *link not fully utilised*

Note: Queue oscillation explains the packet loss phenomena on all sat links we studied – we don't need noise or interference

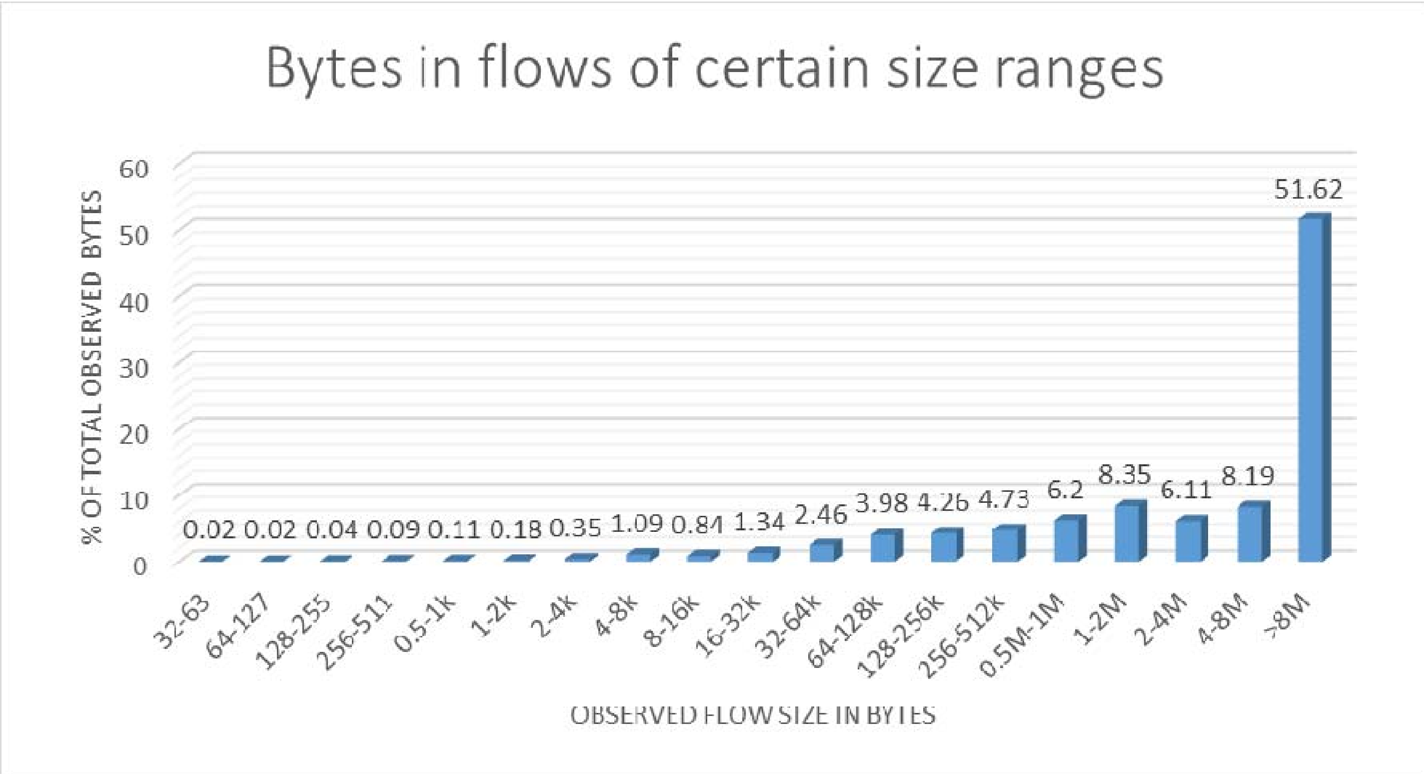
Effect of queue oscillation on TCP flows

- >90% of data bytes end up in TCP connections whose average rate is below 4 Mbps
- But: ISP's don't necessarily see that
- Why?



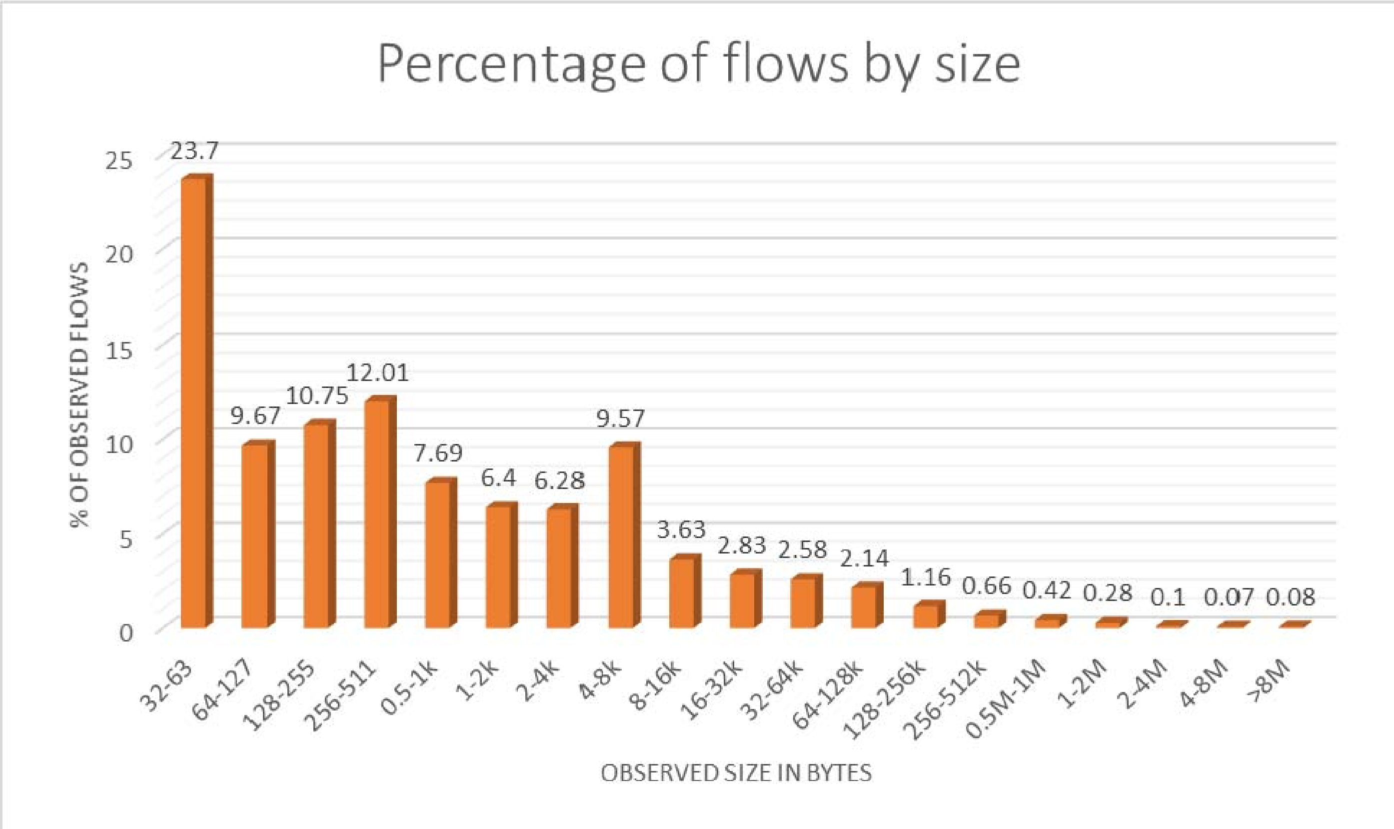
Effect of queue oscillation on TCP flows

- Over half of the bytes are in the largest flows
- Since most bytes experience low data rates…
- But again:
  ISP's don't necessarily see that
- Why?



Why ISP's don't see Queue Oscillation

- Most flows are really short!
- Testing:
  - Ping: short flows
  - Loading web sites: short flows (most elements are small)
- User's can surf and get their mail, (blame server, not ISP)
  - Users don't complain to ISP!
- Short flows don't experience queue oscillation: user experience dominated by RTT delay, not rate



What’s the effect in practice?

- Mail headers and e-mail messages load quickly – but large attachments don’t

- Web browsing pages with only text or small elements is fast – but download of software and larger documents (e.g., PDFs, movies) isn’t

Common misconception

- “The download time depends mainly on the data rate”

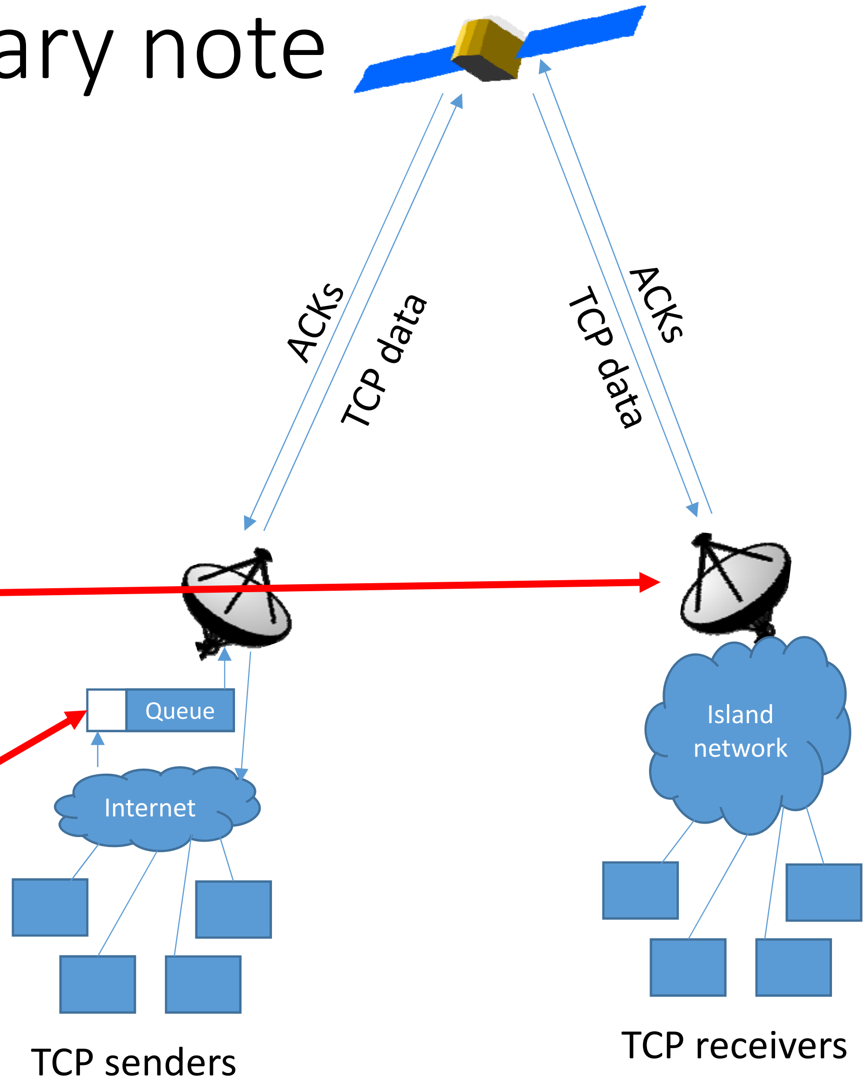
- This is true for large downloads only!

- Example: 160 ms RTT, 300 Mbps sat link, assume no server delay
  - 5 kB download averaging a rate of 4 Mbps takes 170 ms – 94% RTT
    - This is what users experience in web browsing
    - Half the data rate takes this up to 180 ms only
  - 10 MB download averaging a rate of 4 Mbps takes 20.16 s - 0.8% RTT
    - If we only achieve half the data rate here, we need to wait over 40 seconds!

- TCP slow start is meant to let larger flows make better use of the capacity
- This isn’t happening here: shorter flows between 4 and 32kB in size achieve peak data rates around 70% faster than flows of 1MB and more

Possible solutions – preliminary note

- TCP queue oscillation manifests as packet loss
- Communications engineers are generally taught to think of packet loss as cause by low/distorted signal and noise/interference
- In this model, packet loss occurs **at the satellite receiver. This is not what we see here.**
- IP networking people tend to think of packet loss as a congestion phenomenon – packet drops at queues
- Note that in the case of queue oscillation, the losses occurs **at the input queue**, not at the sat receiver



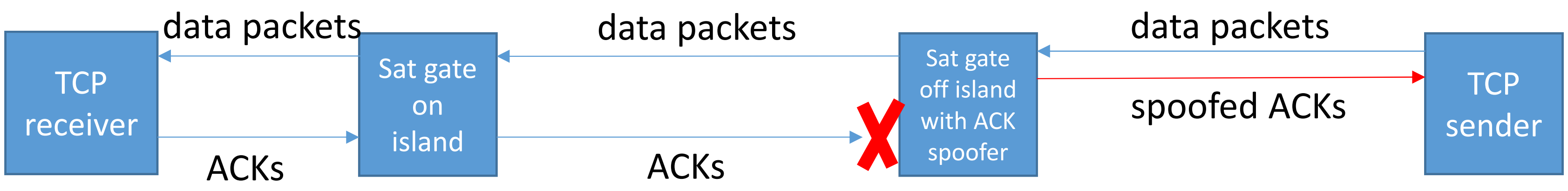
Possible solutions

- Performance Enhancing Proxies (PEPs)
  - Pure ACK spoofers
  - Full connection splitters

- Forward error correction across multiple packets: network coding

ACK-spoofing PEPs

- PEP ACKs incoming data packets to sender
- PEP forwards & caches data packets without modifying sequence numbers
- Absorbs ACKs from receiver or retransmits packets after ACK timeout



- PEP interferes with connection, but doesn't fully split it

Connection-splitting PEPs

- PEP pretends to be the server when dealing with the host that initiates the connection
- PEP terminates the connection and opens a separate connection to the server
- Data between the connections at the PEP travels through a pipe at the application layer

- PEP splits the connection – violates end-to-end principle

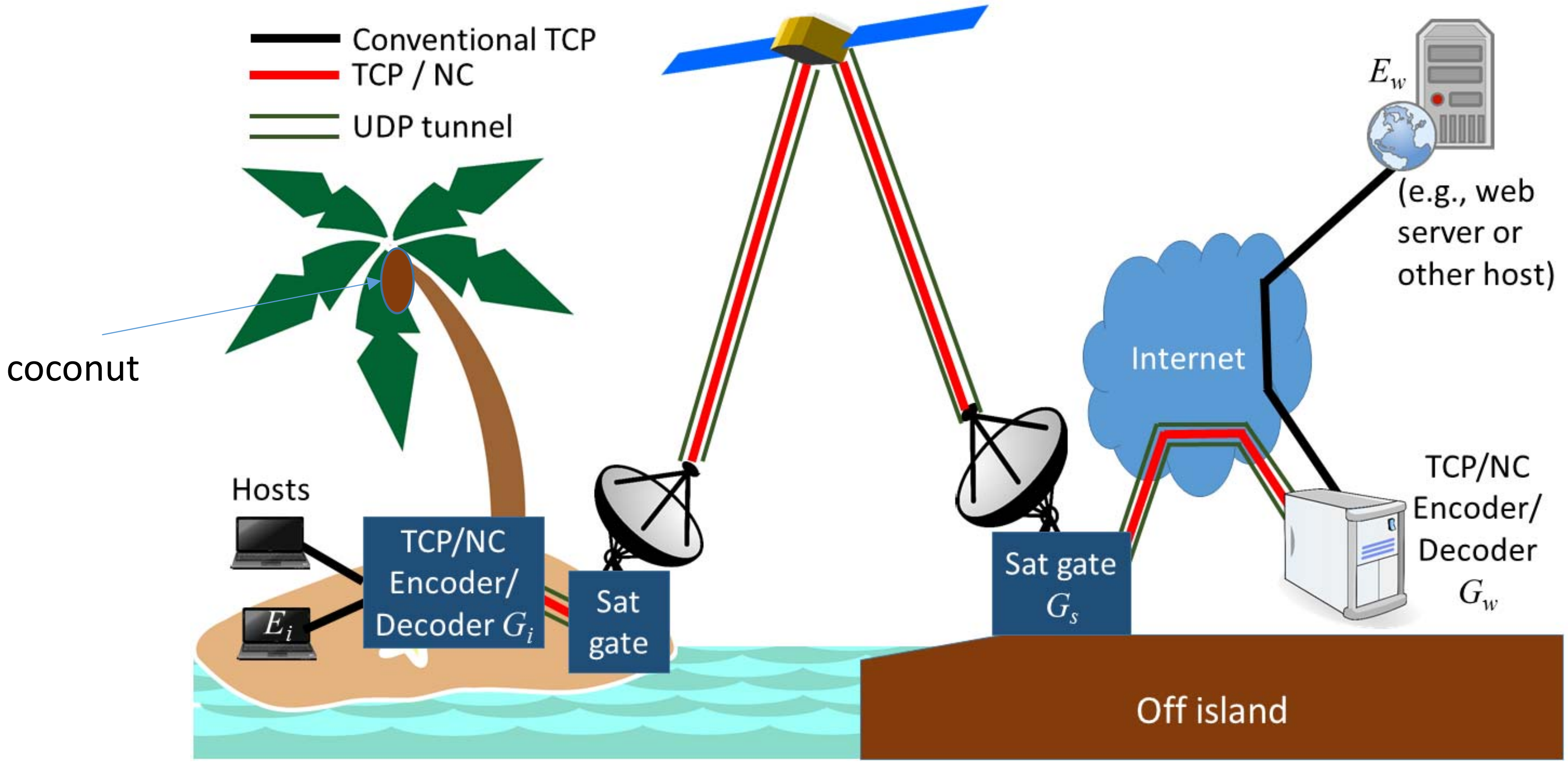
PEPs

- Have been around for a while but either aren’t used in the Pacific, or if used, don’t seem to work that well
- Literature indicates that PEPs work well for a small number of parallel connections, but there are few studies looking at hundreds or thousands of parallel connections
- Connection A potentially still sees a bottleneck with long latency
- Main effect is to bring the RTT down (a little?) for the TCP senders
- Can also have connection-splitting PEPs at both sat gates and use TCP variants for long latencies (Hybla, H-TCP etc.) between sat gates. However, these tend to be aimed at long fat pipes, not long narrow ones.

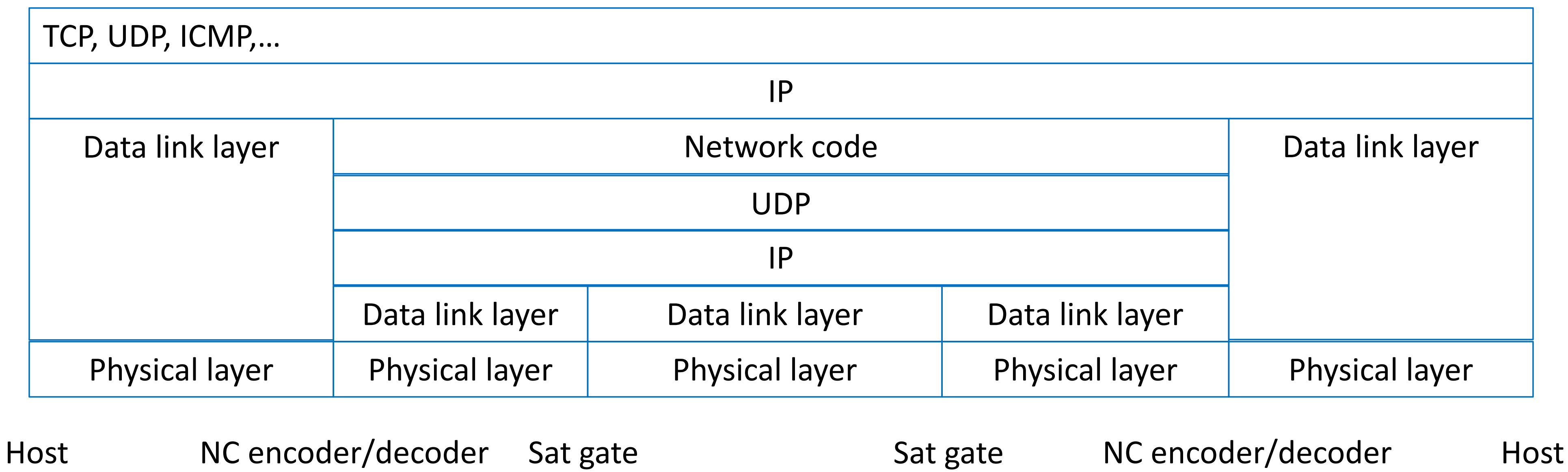
Solution: TCP over network coding (TCP/NC)

- Rethink: Need a way in which we can let the sat gate drop data without causing mayhem in TCP
- Network coding converts g IP packets into N “linear combination packets” (“network encoding”) with N>g
- The decoder at the other end of the satellite link can recover the original g data packets from *any* g out of the N combination packets if they are linearly independent
- Required “overhead” N-g tends to vary slowly in practice – can make this adaptive with feedback

TCP/NC tunnel setup



TCP/NC protocol stack

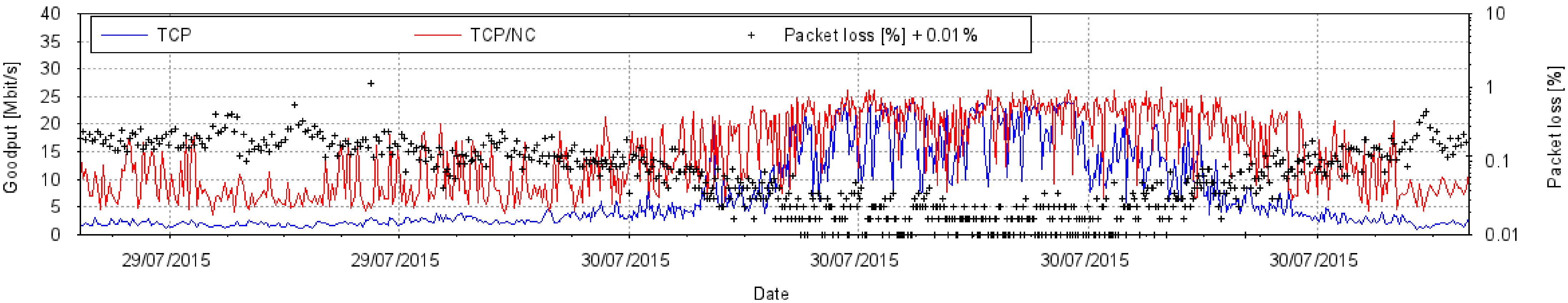


Benefit

- Sat gate can drop up to N-g of the packets without TCP packet loss
- Amount of data going across the sat link is either almost the same as unencoded, or takes up what would have been spare capacity anyway
- Receiver almost never has to wait for missing data – TCP can communicate faster
- Technical effort (cost) involved is lower than the equivalent in extra satellite bandwidth or a cable
- End user don’t need to upgrade their computers
- Larger end users can use network coded TCP for their networks without involving their ISP

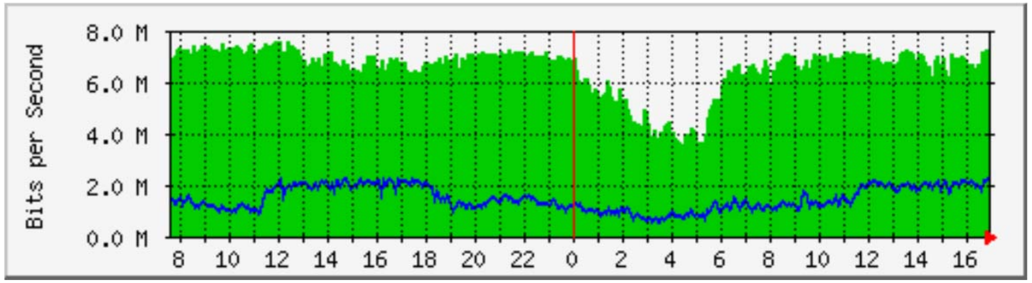
Rarotonga

- O3b satellite connection
- Typical peak time link utilisation around 50%
- TCP/NC encoders/decoders in Avarua and Auckland
- g=30, overhead variable based on feedback
- TCP/NC running alongside standard TCP
    - Would probably need less overhead if all connections were coded



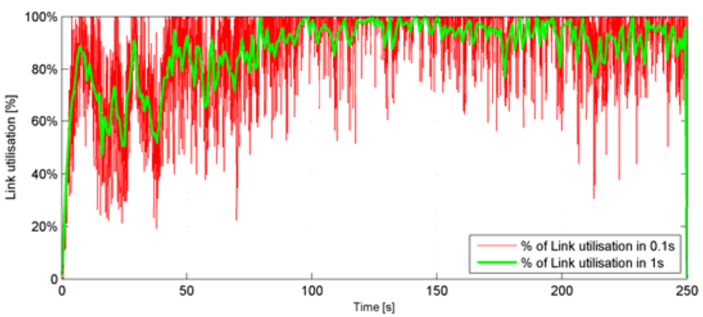
Niue scenario

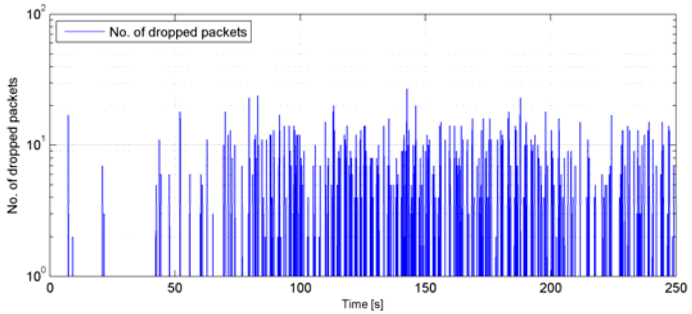
- Connection via geostationary satellite with 8 Mbps downlink
- Very high link utilisation – sustained use of around 7.4 - 7.6 Mbps during the day



Graphic courtesy
Internet Niue

- When sending data to Niue, we see some packet loss
  - Queue overflows but never drains!
- Almost all of the traffic on the link is goodput
  - Can simulate this quite well, too:

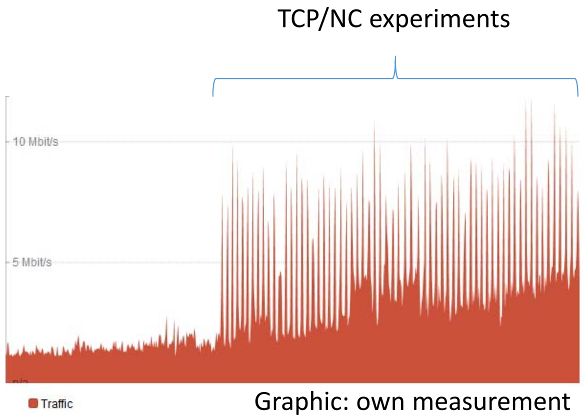




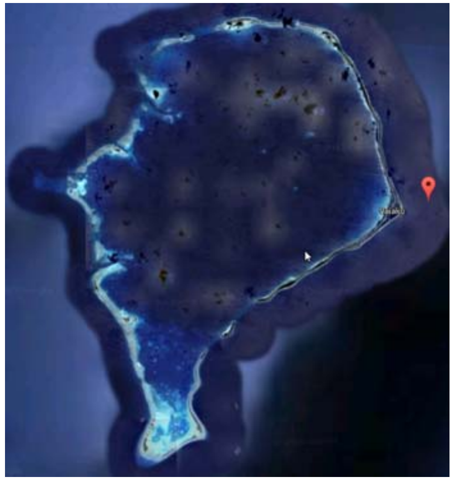
- Non-adaptive TCP/NC buys us goodput of over 2 Mbps on a single connection – at the expense of other traffic!

Funafuti, Tuvalu

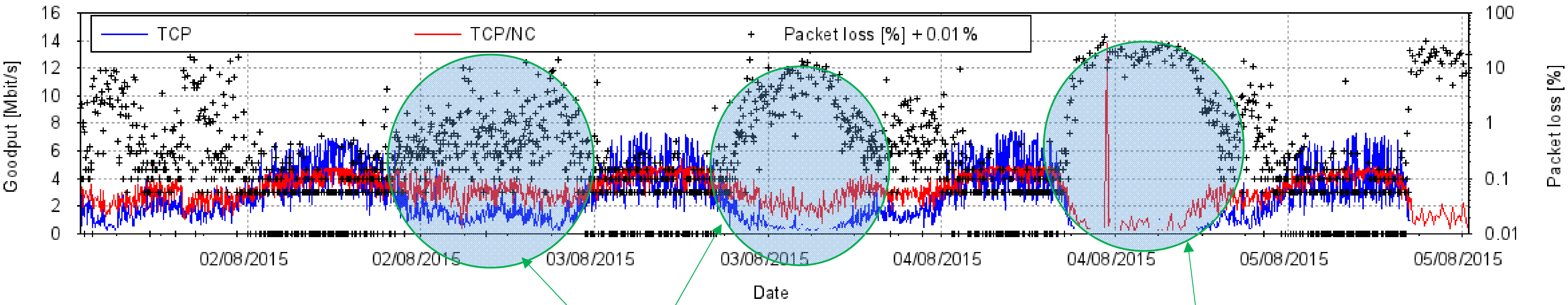
- GEO downlink (~16 Mbps)
- TCP supported by SilverPeak accelerator
- Very low link utilisation below 25%



Graphic: own measurement



- TCP/NC (g=30, adaptive) does not use the SilverPeak



TCP/NC maintains steady goodput with 1-10% packet loss

TCP/NC can complete some downloads even with >10% packet loss

Implementation pathway

- Hardware & infrastructure
  - Off-island
  - On-island
- Software
- Networking considerations
- Transition

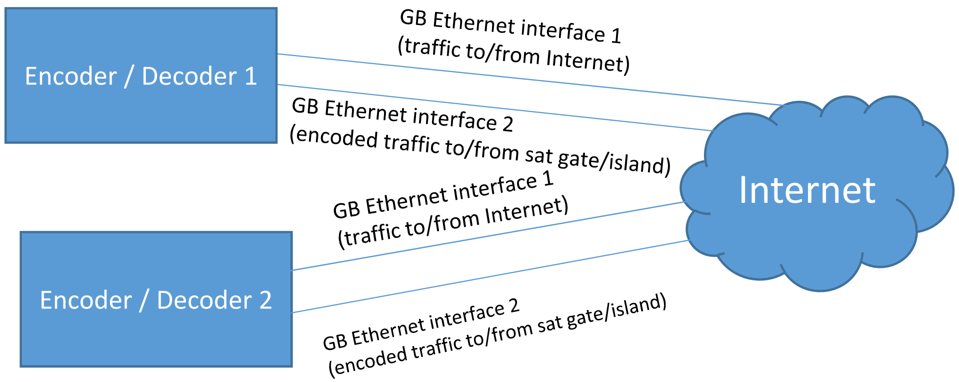
Hardware and infrastructure – off-island

- Need a minimum of TWO encoder/decoder machines off-island (BGP gateway requirement for an AS) in TWO different AS

- Each of these can be a standard standalone server with TWO network interfaces each, running Debian or Ubuntu
  - Most of the computation offshore is encoding, which is less computationally intense than decoding
  - Our machine in Auckland is a Dell PowerEdge 320 with Intel Xeon Processor E5-2420 v2 2.20GHz – as a guide one of these should be powerful enough for sat connections up to several hundred Mbps

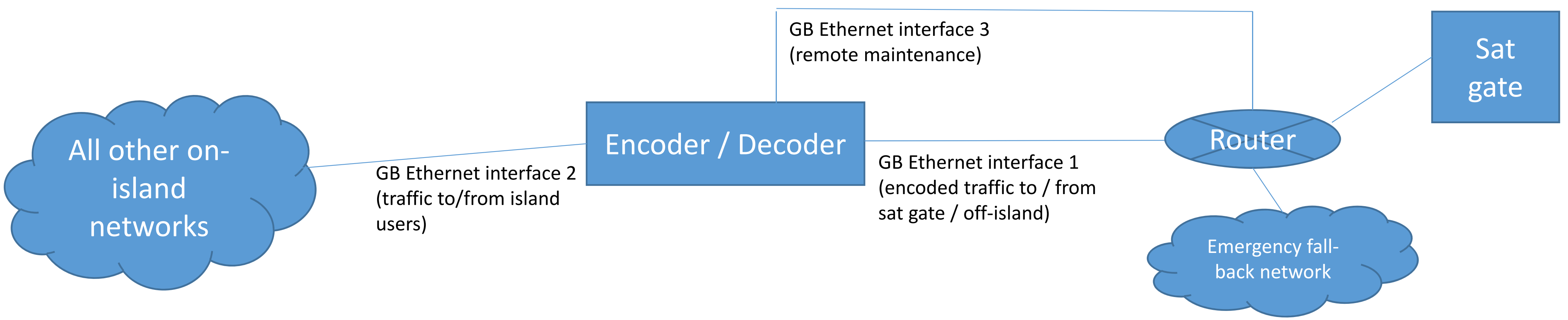
- Should be in a data centre / centres. Best positioned closest to where most large-file traffic comes from (e.g., NZ, California)

- The two network interfaces on each encoder / decoder machine need to be in different subnets



Hardware and infrastructure on-island

- Requires minimum of one encoder/decoder machine (but two for redundancy would be good)
- Encoder/decoder machine(s) should have THREE GB Ethernet interfaces
  - One for tunnel, one to span local network with decoded traffic, one for remote maintenance
- Encoder/decoder machines should sit off the first router after the sat gate
  - Router needs to provide two subnets for the tunnel & remote maintenance interfaces (these can be /30 subnets)
  - One router port faces the sat gate, of course
  - Further router ports are for emergency communication only, in case there are problems with the encoder / decoder



Software

- Will need to refer you to Steinwurf for pricing options
- Current version of software supports two tunnels (one to / from each encoder / decoder on the other side of the link)
- We consider the software (kernel module) very stable – has not crashed on us yet

Networking considerations

- Current (experimental): On-island encoder / decoder spans a /29 subnet of the University of Auckland's 130.216.0.0/16 AS
    - This means we can get away with a single encoder / decoder there
    - Supports only 6 clients (subnet has only 6 spare addresses)

- Production:
    - Would need an AS large enough to accommodate all users (AS "A")
        - Gateways for this AS are the encoder / decoder machines off island – so whoever hosts these needs to advertise the AS via BGP, and their upstream providers need to support this
    - Would need a subnet outside the AS to accommodate sat gate, router, tunnel interface, maintenance interface and emergency fall-back network (AS "B")
        - A /24 is probably more than sufficient for this AS
        - Could belong to the satellite provider
        - Gateways for this AS could be the existing gateways for the current on-island network

Transition (simplified concept)

- Step 1: Get a new /24 AS for AS “B”
- Step 2: Build AS “B” with router and encoder / decoder and configure it such that all it needs is to have its gateways advertised in BGP
- Step 3: Configure the existing gateways to also advertise AS “B”
- Step 4: Configure encoder / decoder machine(s) on-island to act as gateway to the existing on-island, which is part of AS “A”. Use the island-facing IP address of the current router but don’t connect the network yet.
- Step 5: Configure the encoder / decoder machines offshore to act as gateways to AS “A”
- Step 6: Activate tunnel
- Step 7: Advertise off-shore encoder / decoder machines as BGP gateways for AS “A”
- Step 8: Once traffic starts arriving through these, switch on-island network over from old router to on-island encoder / decoder.

Conclusions

- TCP/NC works well under low to moderate queue oscillation – a common daytime scenario in Pacific Islands
- Not so much a matter of "How many times faster?" but more of "How much of my bandwidth can I claw back?"
- Noticeable benefit in Rarotonga and Tuvalu
- Niue simply doesn't have enough bandwidth deployed – TCP/NC gains there squeeze out standard TCP
- No benefit in low demand conditions (Aitutaki in mid-2015)
- Transition requires a bit of network planning

Open questions and progress

- Links shared with legacy TCP cause burst errors that necessitate high NC overhead (high M). What would happen if *all* traffic to an island were encoded? Could we get away with less overhead?

- Current work: Simulating satellite connections with and without TCP/NC and PEPs

- Last but not least: Can we beat the throughput of the coconut telegraph?

Thank you!

Project partners, collaborators and funders

- Aalborg University, Denmark
- Bluesky, Telecom Cook Islands Ltd.
- CAIDA, University of California San Diego and San Diego Supercomputer Center
- Internet Niue
- Internet NZ
- Information Society Innovation Fund (through APNIC)
- Massachusetts Institute of Technology
- Pacific Island Chapter of the Internet Society (PICISOC)
- Steinwurf ApS, Denmark
- Tuvalu Communication Corporation
- University of Auckland