IP routing and QoS in satellite constellations

Networking seminar, INRIA Sophia-Antipolis
Salle Fermat 2, Friday, December 4, 1998, 10:30

Lloyd Wood (L.Wood@surrey.ac.uk)
http://www.ee.surrey.ac.uk/Personal/L.Wood/

Centre for Communication Systems Research,
University of Surrey
http://www.ee.surrey.ac.uk/CCSR/

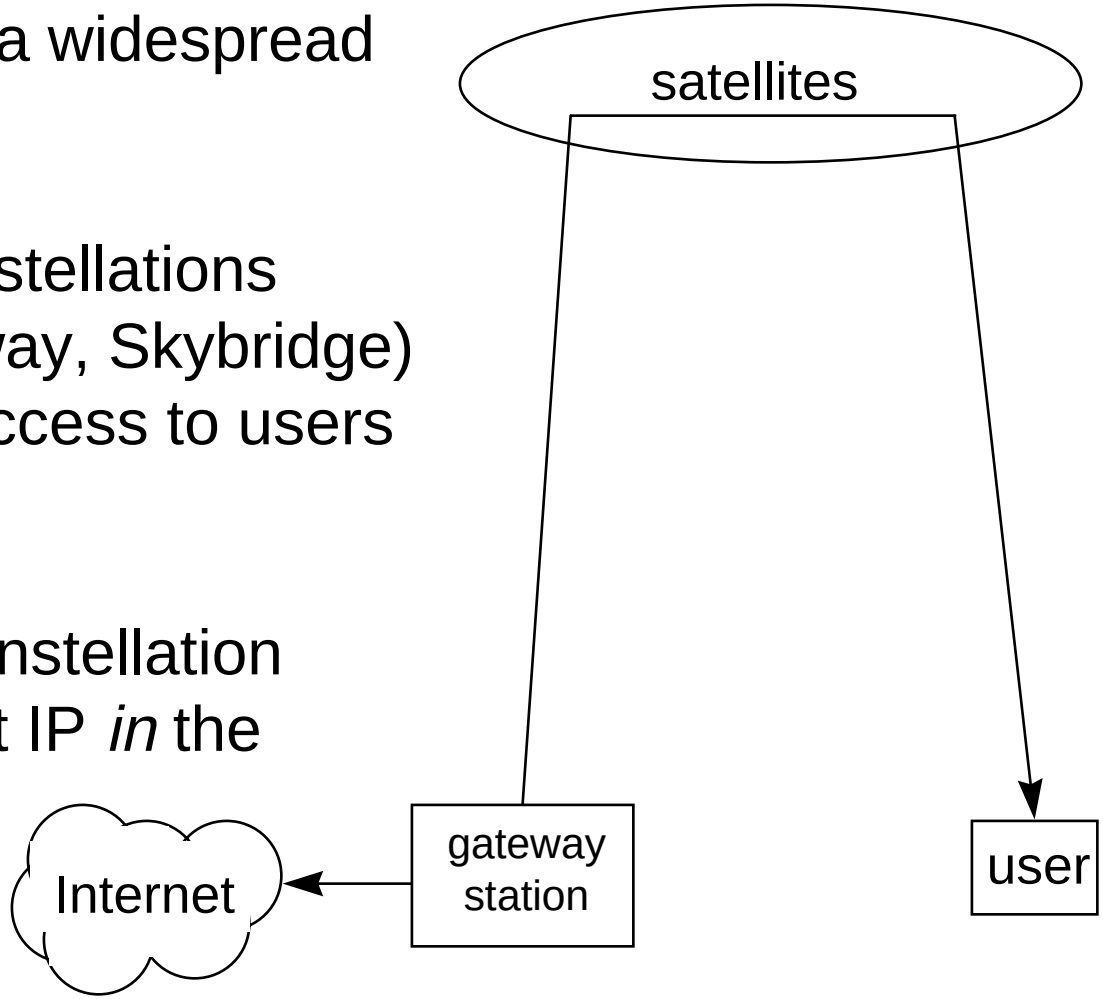
UniS

Outline of scenario

IP (Internet Protocol) is a widespread communication method.

Broadband satellite constellations (e.g. Teledesic, Spaceway, Skybridge) give wireless network access to users in remote areas

Support for IP *by* the constellation is desirable! What about IP *in* the constellation?



Why support IP routing in the constellation?

Advantages:

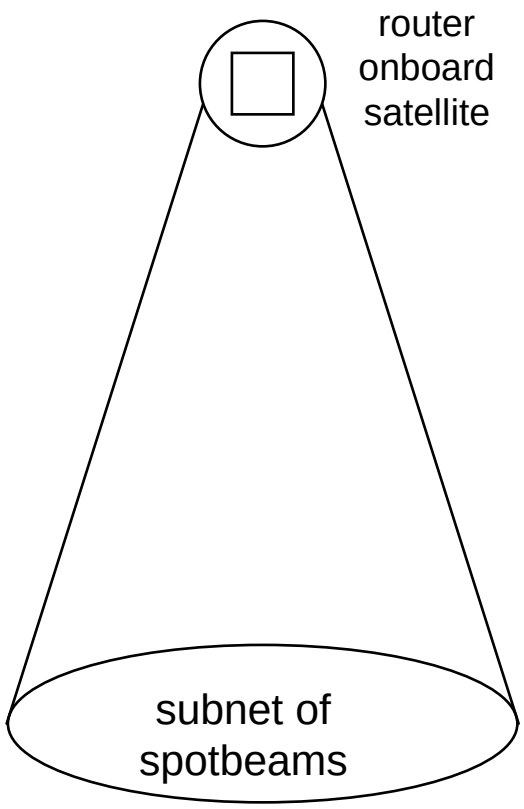
A constellation with intersatellite links forms a mesh network; packetized protocols do well in meshes.

Easy support for IP multicast; better than trying to map IP multicast to a tunnelling layer (e.g. to ATM)

Native support for IP QoS definitions

Simplicity is its own reward!

possible scenario:



But there are disadvantages too:

Non-GEO satellites move. Routing tables move too, either with the satellite (ground renumbers) or against the satellite (the ‘virtual node’/earth-fixed-cell concepts).

Satellite operators can’t make money from routing updates! Advertising updates to rest of Internet is undesirable. Learning about rest of Internet is undesirable for a satellite.

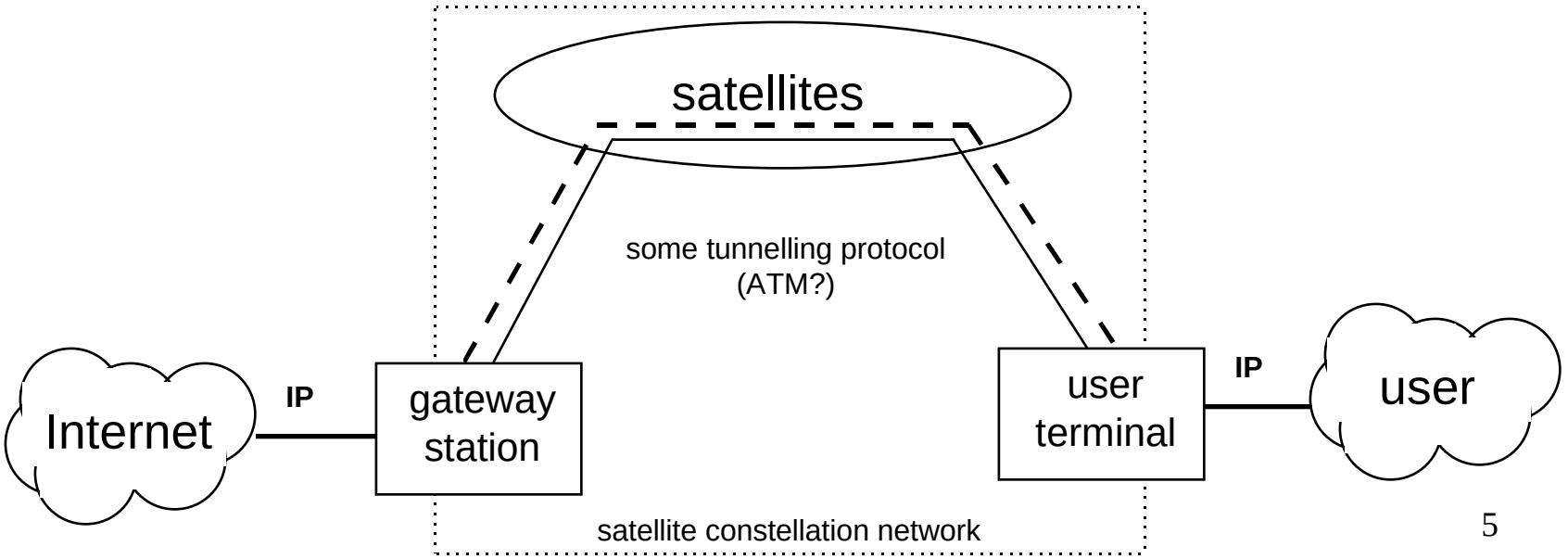
Routing tables grow over time as the Internet grows. You can’t add memory to a satellite in orbit easily...

Satellite on-board processing not as powerful as ground processing - power, radiation constraints.

We could tunnel...

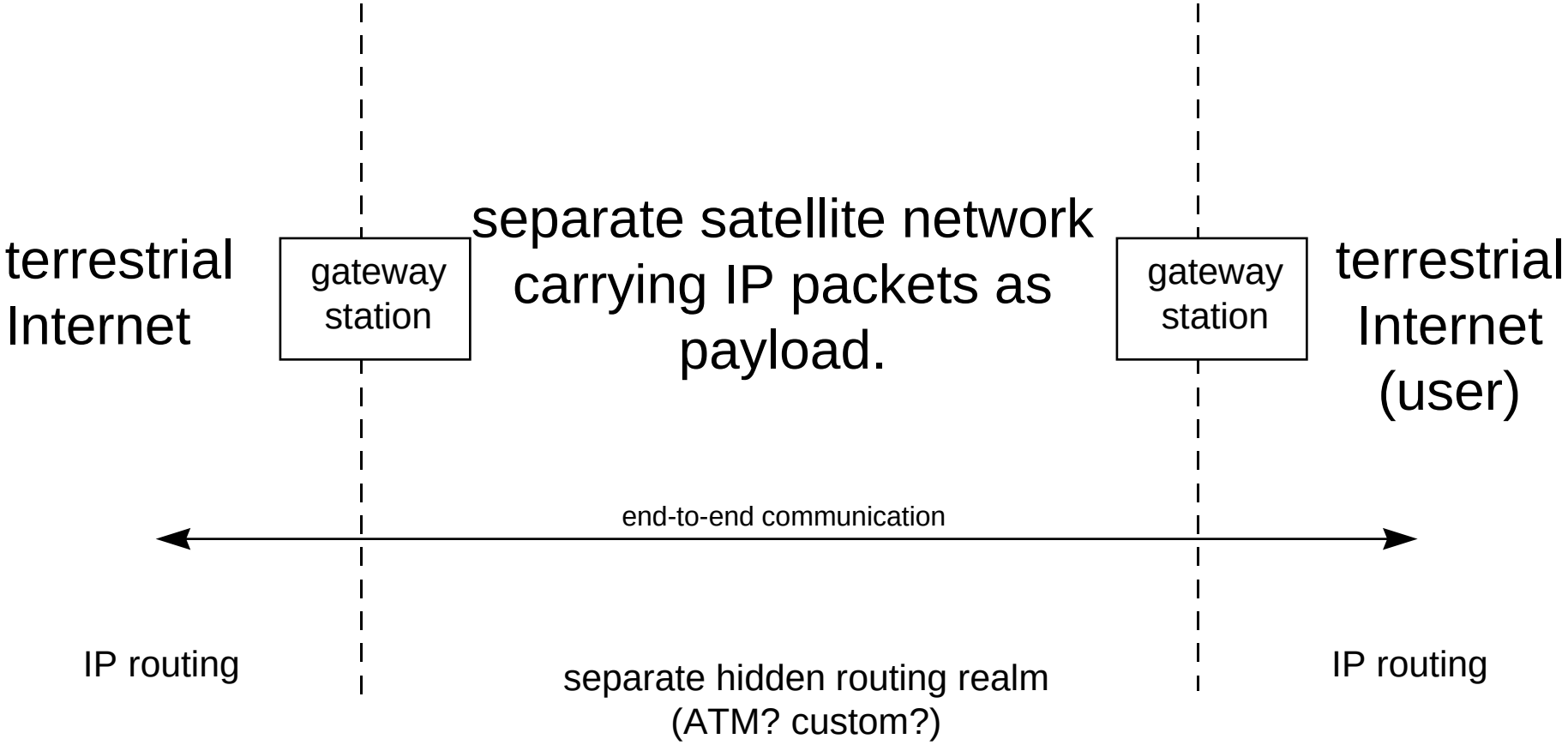
Here the constellation has its own network layer; IP packets are encapsulated in this layer, so routing advertisements from outside and tables storing information on external networks aren't necessary.

Great for end-to-end communication, but you need explicit support for multicast and QoS requirements in each tunnelling node.



From a routing perspective

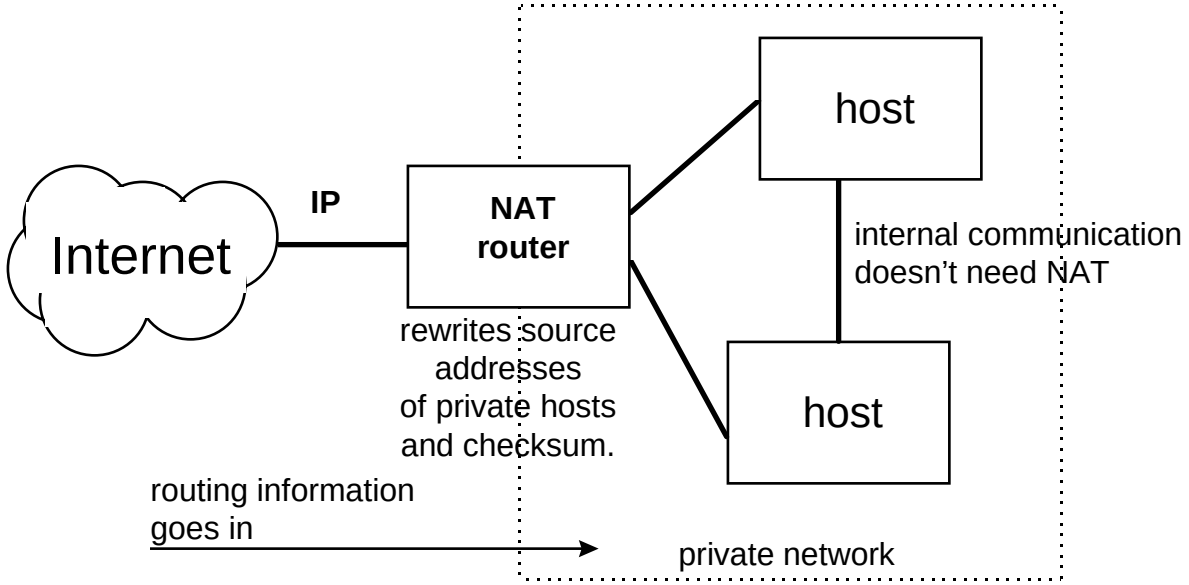
Tunnelling really gives you two separate network layers using separate routing information:



We can get a separate routing realm by using

Network Address Translation (NAT)

increasingly used technique; currently an IETF working group documenting current practice, standardising terminology.



Users can see out; receive routing advertisements from Internet
Internet can't see in; local routing advertisements not propagated beyond gateway - traditional NAT. (bidirectional NAT adds DNS)

NAT in a constellation

NAT used by *private networks* for hiding topology changes or for number reuse. Also for firewalls.

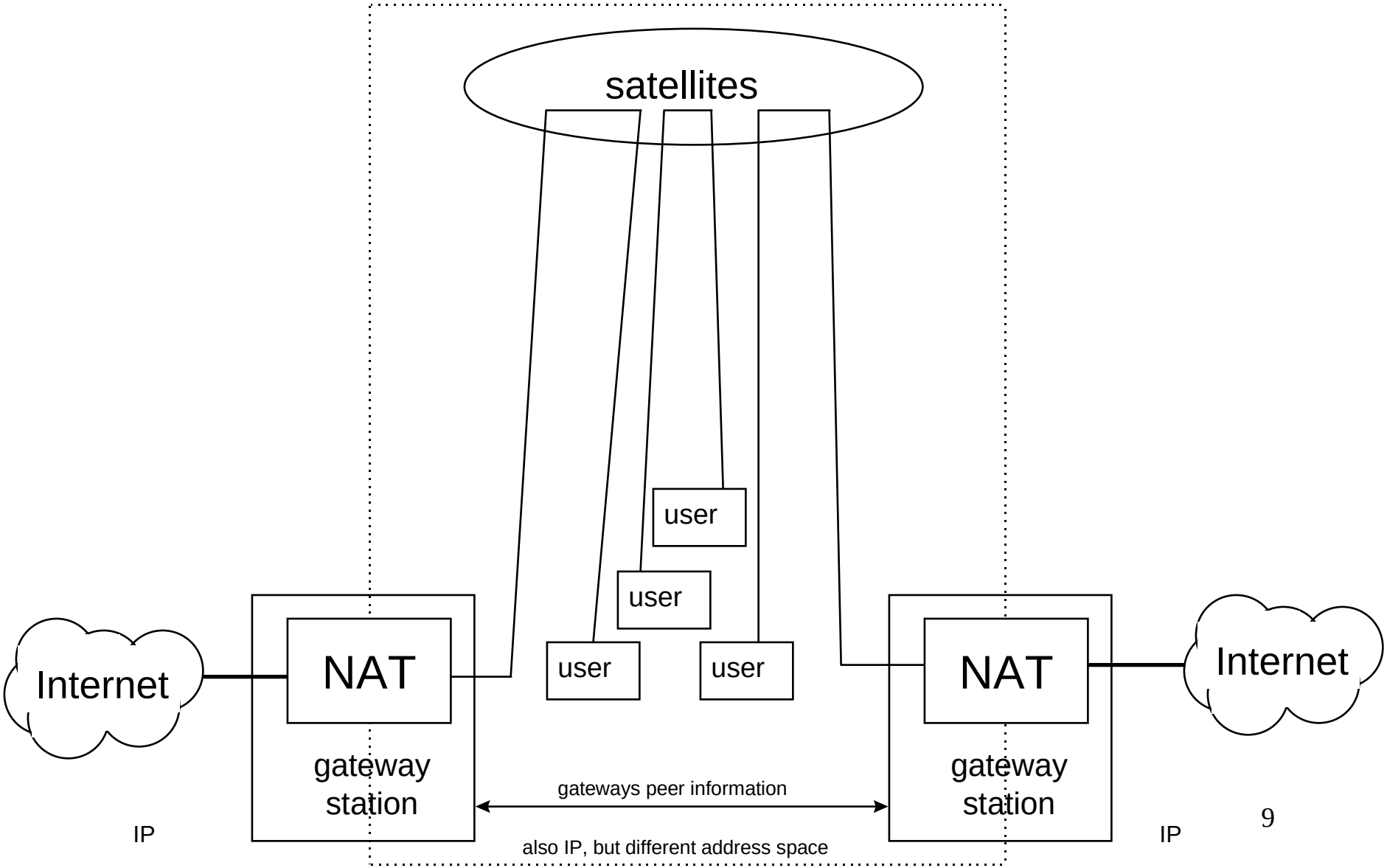
The satellite constellation *is a private network*, although it effectively provides backbone access to its users.

The satellites move; routing information changes; it's state we don't want to maintain.

We don't want routing information coming in *or* out, and we want to pretend that the users are fixed.

We replace source and destination addresses both ways in the NAT router - this is **twice NAT**.

Gateways keep track of satellites; they command and control the constellation and users. They're ideally placed to do NAT.



Multicast and NAT

Multicast is for efficient use of existing capacity; should interest satellite operators since it increases capacity reuse and thus revenue from infrastructure.

end-to-end address locations are semantically overloaded to act as identifiers as well. This is unfortunate from a mobility viewpoint, but made sense to designers of fixed networks.

Multicast is already abstracted to a logical address (Class D addressing); identity and location are separate.

We only need NAT for end-to-end unicast communication - we are unlikely to have to touch multicast. Preferable to the tunnelling approach!

Address association

Give each gateway a block of addresses internally. Assign each address on demand as the translated address of the terrestrial host - routing to gateway is simpler; could mask to establish block/gateway.

Give each gateway a block of addresses externally. Assign a fixed address as the translated address of each satellite user - routing to gateway is simpler.

Terrestrial host thinks it’s talking to the gateway.
Satellite user thinks it’s talking to the gateway.

Multihomed satellite users might need additional NAT...
... but if you’ve got a terrestrial connection, you’re less likely to be using satellite.

Problems with NAT

Conflicts with DNS, but a proper DNS implementation in the gateways works around this.

TCP/IP applications aren't properly layered, and will pass on additional host information (e.g. ftp). You need to rewrite more of the packet than just header addresses and checksum in an *application-level gateway (ALG)*.

Conflicts with IPSec. This may not be a problem from a satellite operator viewpoint; a problem for virtual private networks?

Breaks fragmentation, since only the first fragment has source and destination address - rest are assigned fragmentation IDs. Use gateways to track host/port/frag IDs, or rely on Path MTU Discovery (strongly recommended in an IPv6 world).

The tradeoffs

Engineering NAT and adding tweaks on a per-protocol basis is likely to be easier than engineering a tunnelling layer to add support for each new protocol...

...especially since the tweaking is done on the ground in in the NAT gateways, and not onboard tunnelling satellites.

QoS considerations

NAT breaks RSVP Integrity Objects - but integrated services and per-route guarantees aren't much use when you have a variable-delay varying path across a constellation!

Let the gateways handle RSVP and integrated services, and map to best-effort packetized *differentiated services*.

(Differentiated services - another IETF workgroup with a lot of drafts right now.)

A satellite constellation is homogenous - one set of per-hop behaviours, one domain, one realm for diff-serv; complexity at the edges in the gateways. It's a *simplified* differentiated services implementation.

To summarize...

Tunnelling fakes a shared *protocol* at the edges of your network.

Network Address Translation fakes a shared *address space* at the edges of your network.

Different sets of tradeoffs are involved; it all comes down to the amount of engineering implementation you're willing to do.

NAT lets you engineer protocol support at the edges in the ground, not routing support in the middle in your satellites. For an IP-centric constellation, it's worth considering.