NATx4 Port Allocation and Logging

[Cheng]	draft-cheng-behave-nat44-pre-allocated-ports-01
[Tsou]	draft-tsou-behave-natx4-log-reduction-02
[Durand]	draft-durand-server-logging-recommendations-00

Note Nokia-Siemens IPR declaration on [Tsou]

[durand]

- Recommends logging of source port and address and timestamp at server as well as other information
- Complementary to the other two drafts, won't be mentioned further

[Cheng] vs. [Tsou]

- [Cheng] and [Tsou] describe schemes for allocating ports at the NAT44 in blocks, to reduce the logging volume
  - Log only when blocks allocated, rather than each time a port is allocated
  - Both allow randomization
- Primary differences between [Cheng] and [Tsou]:
  - [Cheng] puts a static per-subscriber limit on total ports allocated. [Tsou] allocates blocks without limit as required.
  - [Tsou] considers block deallocation. [Cheng] does not mention it.
- [Cheng] (maybe) and [Tsou] have issue of garbage collection (return of unused ports to common pool)
  - Tradeoff between randomization, clearance of little-used blocks
  - How soon to free block where all ports appear to be idle

Static port management

- **Pros:**
  - simple to understand
  - DHCP-style logs
    - Log initial port range and be done.
- **Cons:**
  - Security
    - port randomization entropy is reduced to bucket size
    - Easy to mount attacks if bucket is small
  - Operation
    - No mechanism to extend bucket
    - Complex failures when port range is exhausted
    - Usually leads to very large buckets
      ⍰ sub-optimal use of IP address
      5000 ports/user => 10 user/IP address

Dynamic port management

- Pros:
  - Large statistic multiplexing
  - All users: Average 5 port/user
    - 10,000 users/IP address
  - Active users only: Average 100 ports/user
    - 650 users/IP address
- Cons
  - Need to log each NAT binding
  - 1 binding: 16 bytes, 2000 cnx/user/day, 6 month logs, 1,000,000 users = 5.6 Terabyte of data
  - 1 binding: 20 bytes, 10000 cnx/user/day, 2 year logs,
    1,000,000 users = 150 Terabyte of data

⍰ Lot of data to store/archive/search

Hybrid port management: buckets

Solution 1

- Allocate ports in small buckets of random ports, say 20 at a time
  - When port is released, return it to free pool
- Log creation of bucket, not each flow
- Divide log volume & messages by 20

Pros:

- Better logs
- Preserve randomization
- Small impact on IP utilization ratio

Cons:

- Still lot of logs
- More complexity to manage buckets

Hybrid port management: static + dynamic buckets

Solution 2

- Based on solution 1
- 1st bucket is “special”:
  - Larger (eg 200 ports)
  - Released ports are put back in the bucket to be reused by the same user
- Other buckets works the same as solution 1
- Create a static random set of ports per user, with possibility to add new ports as needed

static + dynamic buckets analysis

Security

- Initial bucket is made of random ports
  - But an attacker could discover them
- Subsequent buckets are totally random

Operation

- Guarantees a minimum of ports per user
- Extend dynamically that range if/when needed
- Logs reduced to zero for users who stay in their initial bucket
- Multiplexing: about 250 users per IP address

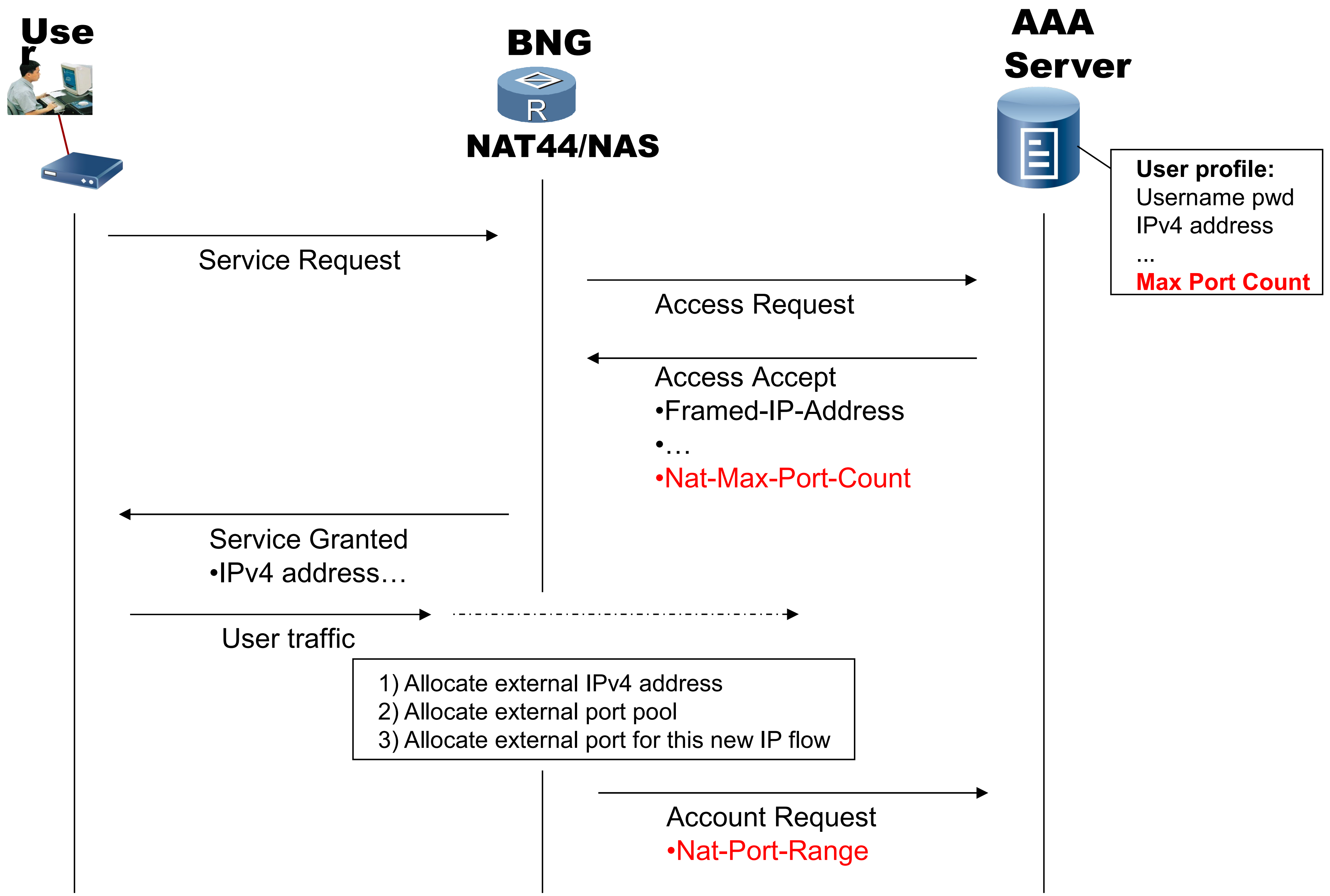
Conclusion

- Port management offers a trade-off:
  log size vs address oversubscription ratio
  - Static management:
    - No logs, low over-subscription ratio
  - Dynamic management:
    - High volume of logs, high over-subscription ratio
  - Hybrid methods:
    - Medium to small volume of logs, medium over-subscription ratio

Backup Slides

Details of Proposals

[Cheng] Message Flow



Use
r

BNG

NAT44/NAS

AAA
Server

User profile:
Username pwd
IPv4 address
...
Max Port Count

Service Request

Access Request

Access Accept
•Framed-IP-Address
•…
•Nat-Max-Port-Count

Service Granted
•IPv4 address…

User traffic

1) Allocate external IPv4 address
2) Allocate external port pool
3) Allocate external port for this new IP flow

Account Request
•Nat-Port-Range