Document: draft-cheshire-nat-pmp-02.txt
Internet-Draft
Category: Standards Track
Expires 14th March 2007

Stuart Cheshire
Marc Krochmal
Apple Computer, Inc.
Kiren Sekar
Sharpcast, Inc.
14th September 2006

NAT Port Mapping Protocol (NAT-PMP)

<draft-cheshire-nat-pmp-02.txt>

Status of this Memo

Abstract

This document describes a protocol for automating the process of creating Network Address Translation (NAT) port mappings. Included in the protocol is a method for retrieving the public IP address of a NAT gateway, thus allowing a client to make this public IP address and port number known to peers that may wish to communicate with it. This protocol is implemented in current Apple products including Mac OS X, Bonjour for Windows, and AirPort wireless base stations.

1. Introduction

Network Address Translation (NAT) is a method of sharing one public internet address with a number of devices. This document is focused on what "IP Network Address Translator (NAT) Terminology and Considerations" [RFC 2663] calls "NAPTs" (Network Address/Port Translators). A full description of NAT is beyond the scope of this document. The following brief overview will cover the aspects relevant to this port mapping protocol. For more information on NAT, see "Traditional IP Network Address Translator" [RFC 3022].

NATs have one or more public IP addresses. A private network is set up behind the NAT. Devices behind the NAT are assigned private addresses and the private address of the NAT device is used as the gateway.

When a packet from any device behind the NAT is sent to an address on the public internet, the packet first passes through the NAT box. The NAT box looks at the source port and address. In some cases, a NAT will also keep track of the destination port and address. The NAT then creates a mapping from the private address and private port to a public address and public port if a mapping does not already exist. The NAT box replaces the private address and port number in the packet with the public entries from the mapping and sends the packet on to the next gateway.

When a packet from any address on the internet is received on the NAT's public side, the NAT will look up the destination port (public port) in the list of mappings. If an entry is found, it will contain the private address and port that the packet should be sent to. The NAT gateway will then rewrite the destination address and port with those from the mapping. The packet will then be forwarded to the new destination addresses. If the packet did not match any mapping, the packet will most likely be dropped. Various NATs implement different strategies to handle this. The important thing to note is that if there is no mapping, the NAT does not know which private address the packet should be sent to.

Mappings are usually created automatically as a result of observing outbound traffic. There are a few exceptions. Some NATs may allow manually-created permanent mappings that map a public port to a specific private IP address and port. Such a mapping allows incoming connections to the device with that private address. Some NATs also implement a default mapping where any inbound traffic that does not match a mapping will always be forwarded to a specific private address. Both types of mappings are usually set up manually through some configuration tool.

Without these manually-created inbound port mappings, clients behind the NAT would be unable to receive inbound connections, which represents a loss of connectivity when compared to the original

Internet architecture [ETEAISD]. For those who view this loss of connectivity as a bad thing, NAT-PMP allows clients to operate much more like a host directly connected to the unrestricted public Internet, with an unrestricted public IP address. NAT-PMP allows client hosts to communicate with the NAT gateway to request the creation of inbound mappings on demand. Having created a NAT mapping to allow inbound connections, the client can then record its public IP address and public port number in a public registry (e.g. the world-wide Domain Name System) or otherwise make it accessible to peers that wish to communicate with it.

2. Conventions and Terminology Used in this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC 2119].

3. Protocol and Packet Format

NAT Port Mapping Protocol runs over UDP. Every packet starts with an 8 bit version followed by an 8 bit operation code.

This document specifies version 0 of the protocol. Any NAT-PMP gateway implementing this version of the protocol, receiving a packet with a version number other than 0, MUST return result code 1

(Unsupported Version).

Opcodes between 0 and 127 are client requests. Opcodes from 128 to 255 are server responses. Responses always contain a 16 bit result code in network byte order. A result code of zero indicates success. Responses also contain a 32 bit unsigned integer corresponding to the number of seconds since the NAT gateway was rebooted or since its port mapping state was reset.

This protocol SHOULD only be used when the client determines that its primary IPv4 address is in one of the private IP address ranges defined in "Address Allocation for Private Internets" [RFC 1918]. This includes the address ranges 10/8, 172.16/12, and 192.168/16.

Clients always send their Port Mapping Protocol requests to their default gateway, as learned via DHCP [RFC 2131], or similar means. This protocol is designed for small home networks, with a single logical link (subnet) where the client's default gateway is also the NAT translator for that network. For more complicated networks where the NAT translator is some device other than the client's default gateway, this protocol is not appropriate.

3.1 Requests and Responses

NAT gateways are often low-cost devices, with limited memory and CPU speed. For this reason, to avoid making excessive demands on the NAT gateway, clients machines SHOULD NOT issue multiple requests simultaneously in parallel. If a client needs to perform multiple requests (e.g. on boot, wake from sleep, network connection, etc.) it SHOULD queue them and issue them serially one at a time. Once the NAT gateway responds to one request the client machine may issue the next. In the case of a fast NAT gateway, the client may be able to complete requests at a rate of hundreds per second. In the case of a slow NAT gateway that takes perhaps half a second to respond to a NAT-PMP request, the client SHOULD respect this and allow the NAT gateway to operate at the pace it can manage, and not overload it by issuing requests faster than the rate it's answering them.

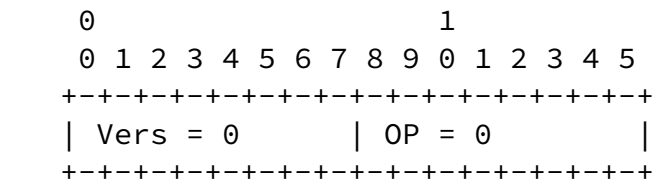
To determine the puclic IP address or request a port mapping, a NAT-PMP client sends its request packet to port 5351 of its configured gateway address, and waits 250ms for a response. If no

NAT-PMP response is received from the gateway after 250ms, the client retransmits its request and waits 500ms. The client SHOULD repeat this process with the interval between attempts doubling each time. If, after sending its 9th attempt (and then waiting for 64 seconds), the client has still received no response, then it SHOULD conclude that this gateway does not support NAT Port Mapping Protocol and MAY log an error message indicating this fact. In addition, if the NAT-PMP client receives an "ICMP Port Unreachable" message from the gateway for port 5351 then it can skip any remaining retransmissions and conclude immediately that the gateway does not support NAT-PMP.

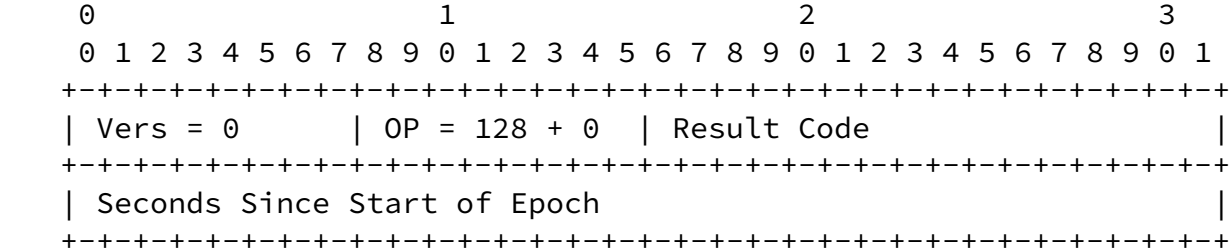
As a performance optimization the client MAY record this information and use it to suppress further attempts to use NAT-PMP, but the client should not retain this information for too long. In particular, any event that may indicate a potential change of gateway or a change in gateway configuration (hardware link change indication, change of gateway MAC address, acquisition of new DHCP lease, receipt of NAT-PMP announcement packet from gateway, etc.) should cause the client to discard its previous information regarding the gateway's lack of NAT-PMP support, and send its next NAT-PMP request packet normally.

3.2 Determining the Public Address

To determine the public address, the client behind the NAT sends the following UDP payload to port 5351 of the configured gateway address:



A compatible NAT gateway MUST generate a response with the following format:



```
| Public IP Address (a.b.c.d)                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This response indicates that the NAT gateway implements this version of the protocol and returns the public IP address of the NAT gateway. If the result code is non-zero, the value of Public IP Address is undefined (MUST be set to zero on transmission, and MUST be ignored on reception).

The NAT gateway MUST fill in the "Seconds Since Start of Epoch" field with the time elapsed since its port mapping table was initialized on startup or reset for any other reason (see Section 3.6 "Seconds Since Start of Epoch").

Upon receiving the response packet, the client MUST check the source IP address, and silently discard the packet if the address is not the address of the gateway to which the request was sent.

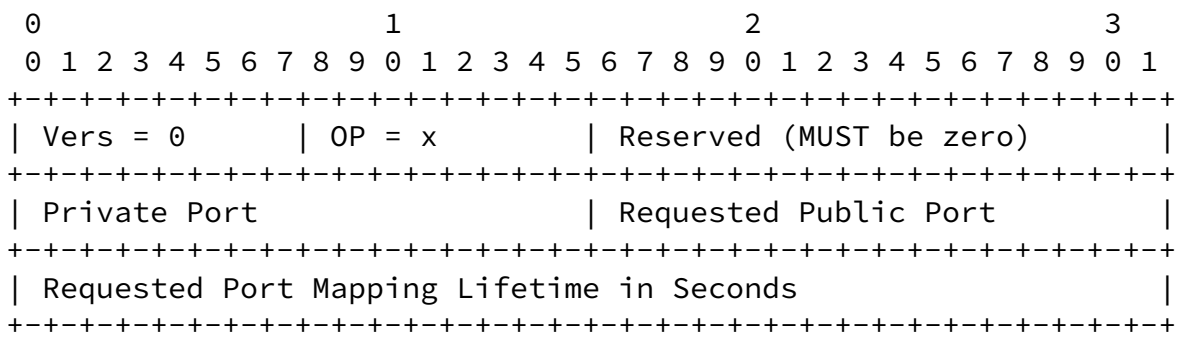
3.2.1 Announcing Address Changes

When the public IP address of the NAT changes, the NAT gateway MUST send a gratuitous response to the link-local multicast address 224.0.0.1, port 5351 with the packet format above to notify clients of the new public IP address. To accommodate packet loss, the NAT gateway SHOULD multicast 10 address change notifications. The interval between the first two notifications SHOULD be 250ms, and the interval between each subsequent notification SHOULD double.

Upon receiving a gratuitous address change announcement packet, the client MUST check the source IP address, and silently discard the packet if the address is not the address of the client's current configured gateway. This is to guard against inadvertent misconfigurations where there may be more than one NAT gateway active on the network.

3.3 Creating a Mapping

To create a mapping, the client sends a UDP packet to port 5351 of the gateway's private IP address with the following format:



Opcodes supported:
1 - Map UDP
2 - Map TCP

The Reserved field MUST be set to zero on transmission and MUST be ignored on reception.

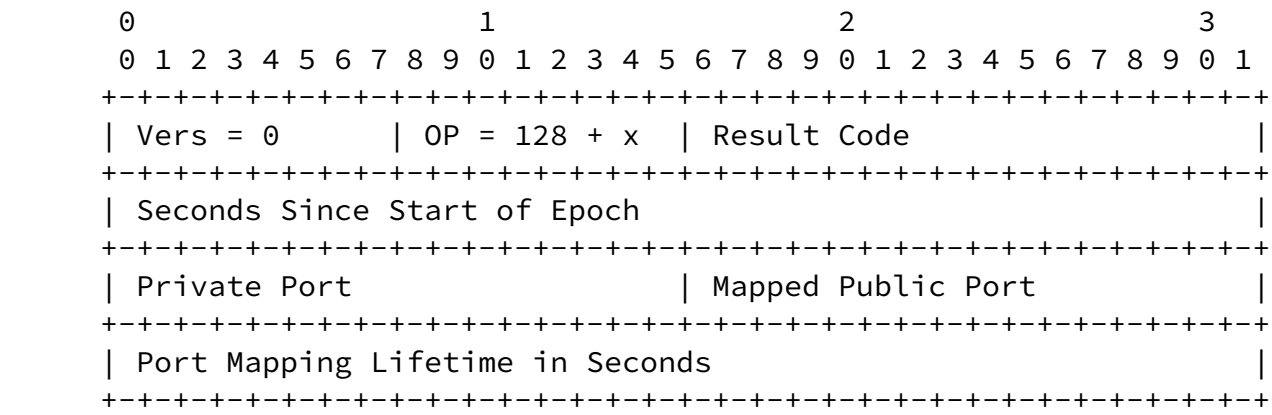
The Private Port is set to the local port on which the client is listening.

The Requested Public Port SHOULD usually be set to the same value as the local Private Port, or zero if the client has no preference for what port is assigned. However, the gateway is not obliged to assign the port requested, and may choose not to, either for policy reasons (e.g. port 80 is reserved and clients may not request it) or because that port has already been assigned to some other client. Because of this, some product developers have questioned the value of having the Requested Public Port field at all. The reason is for failure recovery. Most low-cost home NAT gateways do not record temporary port mappings in persistent storage, so if the gateway crashes or is rebooted, all the mappings are lost. A renewal packet is formatted identically to an initial mapping request packet, except that for renewals the client sets the Requested Public Port field to the port the gateway actually assigned, rather than the port the client originally wanted. When a freshly-rebooted NAT gateway receives a renewal packet from a client, it appears to the gateway just like an ordinary initial request for a port mapping, except that in this case the Requested Public Port is likely to be one that the NAT gateway *is* willing to allocate (it allocated it to this client right before the reboot, so it should presumably be willing to allocate it again).

The RECOMMENDED Port Mapping Lifetime is 3600 seconds.

After sending the port mapping request, the client then waits for the NAT gateway to respond. If after 250ms, the gateway doesn't respond, the client SHOULD re-issue its request as described above in Section 3.1 "Requests and Responses".

The NAT gateway responds with the following packet format:



The 'x' in the OP field MUST match what the client requested. Some NAT gateways are incapable of creating a UDP port mapping without also creating a corresponding TCP port mapping, and vice versa, and these gateways MUST NOT implement NAT Port Mapping Protocol until this deficiency is fixed. A NAT gateway which implements this protocol MUST be able to create TCP-only and UDP-only port mappings.

If a NAT gateway silently creates a pair of mappings for a client that only requested one mapping, then it may expose that client to receiving inbound UDP packets or inbound TCP connection requests that it did not ask for and does not want.

While a NAT gateway MUST NOT automatically create mappings for TCP when the client requests UDP, and vice versa, the NAT gateway MUST reserve the companion port so the same client can choose to map it in the future. For example, if a client requests to map TCP port 80, as long as the client maintains the lease for that TCP port mapping, another client with a different IP address MUST NOT be able to successfully acquire the mapping for UDP port 80.

The client normally requests the public port matching the private port. If that public port is not available, the NAT gateway MUST return a public port that is available or return an error code if no ports are available.

The source address of the packet MUST be used for the private address in the mapping. This protocol is not intended to facilitate one device behind a NAT creating mappings for other devices. If there are legacy devices that require inbound mappings, permanent mappings can be created manually by the administrator, just as they are today.

If a mapping already exists for a given private port on a given local client (whether that mapping was created explicitly using NAT-PMP, implicitly as a result of an outgoing TCP SYN packet, or manually by a human administrator) and that client requests another mapping for the same private port (possibly requesting a different public port) then the mapping request should succeed, returning the already-assigned public port. This is necessary to handle the case where a client requests a mapping with requested public port X, and is granted a mapping with actual public port Y, but the acknowledgement packet gets lost. When the client retransmits its mapping request, it should get back the same positive acknowledgement as was sent (and lost) the first time.

The NAT gateway SHOULD NOT accept mapping requests destined to the NAT gateway's public IP address or received on its public network interface. Only packets received on the private interface(s) with a destination address matching the private address(es) of the NAT gateway should be allowed.

The NAT gateway MUST fill in the "Seconds Since Start of Epoch" field with the time elapsed since its port mapping table was initialized on startup or reset for any other reason (see Section 3.6 "Seconds Since Start of Epoch").

The Port Mapping Lifetime is an unsigned integer in seconds. The NAT gateway MAY reduce the lifetime from what the client requested. The NAT gateway SHOULD NOT offer a lease lifetime greater than that requested by the client.

Upon receiving the response packet, the client MUST check the source IP address, and silently discard the packet if the address is not the address of the gateway to which the request was sent.

The client SHOULD begin trying to renew the mapping halfway to expiry

time, like DHCP. The renewal packet should look exactly the same as a request packet, except that the client SHOULD set the requested public port to what the NAT gateway previously mapped, not what the client originally requested. As described above, this enables the gateway to automatically recover its mapping state after a crash or reboot.

3.4 Destroying a Mapping

A mapping may be destroyed in a variety of ways. If a client fails to renew a mapping, then when its lifetime expires the mapping MUST be automatically deleted. In the common case where the gateway device is a combined DHCP server and NAT gateway, when a client's DHCP address lease expires, the gateway device MAY automatically delete any mappings belonging to that client. Otherwise a new client being assigned the same IP address could receive unexpected inbound

UDP packets or inbound TCP connection requests that it did not ask for and does not want.

A client MAY also send an explicit packet to request deletion of a mapping that is no longer needed. A client requests explicit deletion of a mapping by sending a message to the NAT gateway requesting the mapping, with the Requested Lifetime in Seconds set to 0. The requested public port MUST be set to zero by the client on sending, and MUST be ignored by the gateway on reception.

When a mapping is destroyed successfully as a result of the client explicitly requesting the deletion, the NAT gateway MUST send a response packet which is formatted as defined in section 3.3 "Creating a Mapping". The response MUST contain a result code of 0, the private port as indicated in the deletion request, a public port of 0, and a lifetime of 0. The NAT gateway MUST respond to a request to destroy a mapping that does not exist as if the request were successful. This is because of the case where the acknowledgement is lost, and the client retransmits its request to delete the mapping. In this case the second request to delete the mapping MUST return the same response packet as the first request.

If the deletion request was unsuccessful, the response MUST contain a non-zero result code and the requested mapping; the lifetime is undefined (MUST be set to zero on transmission, and MUST be ignored

on reception). If the client attempts to delete a port mapping which was manually assigned by some kind of configuration tool, the NAT gateway MUST respond with a 'Not Authorized' error, result code 2.

When a mapping is destroyed as a result of its lifetime expiring or for any other reason, if the NAT gateway's internal state indicates that there are still active TCP connections traversing that now-defunct mapping, then the NAT gateway SHOULD send appropriately-constructed TCP RST (reset) packets both to the local client and to the remote peer on the Internet to terminate that TCP connection.

A client can request the explicit deletion of all its UDP or TCP mappings by sending the same deletion request to the NAT gateway with public port, private port, and lifetime set to 0. A client MAY choose to do this when it first acquires a new IP address in order to protect itself from port mappings that were performed by a previous owner of the IP address. After receiving such a deletion request, the gateway MUST delete all its UDP or TCP port mappings (depending on the opcode). The gateway responds to such a deletion request with a response as described above, with the private port set to zero. If the gateway is unable to delete a port mapping, for example, because the mapping was manually configured by the administrator, the gateway MUST still delete as many port mappings as possible, but respond with a non-zero result code. The exact result code to return depends on the cause of the failure. If the gateway is able to successfully delete all port mappings as requested, it MUST respond with a result code of 0.

3.5 Result Codes

Currently defined result codes:
0 - Success
1 - Unsupported Version
2 - Not Authorized/Refused
(e.g. box supports mapping, but user has turned feature off)
3 - Network Failure
(e.g. NAT box itself has not obtained a DHCP lease)
4 - Out of resources
(NAT box cannot create any more mappings at this time)
5 - Unsupported opcode

If the result code is non-zero, the format of the packet following the result code may be truncated. For example, if the client sends a request to the server with an opcode of 17 and the server does not

recognize that opcode, the server SHOULD respond with a message where the opcode is 17 + 128 and the result code is 5 (opcode not supported). Since the server does not understand the format of opcode 17, it may not know what to place after the result code. In some cases, relevant data may follow the opcode to identify the operation that failed. For example, a client may request a mapping but that mapping may fail due to resource exhaustion. The server SHOULD respond with the result code to indicate resource exhaustion (4) followed by the requested port mapping so the client may identify which operation failed.

Clients MUST be able to properly handle result codes not defined in this document. Undefined results codes MUST be treated as fatal errors of the request.

3.6 Seconds Since Start of Epoch

Every packet sent by the NAT gateway includes a "Seconds since start of epoch" field (SSSOE). If the NAT gateway resets or loses the state of its port mapping table, due to reboot, power failure, or any other reason, it MUST reset its epoch time and begin counting SSSOE from 0 again. Whenever a client receives any packet from the NAT gateway, either gratuitously or in response to a client request, the client computes its own conservative estimate of the expected SSSOE value by taking the SSSOE value in the last packet it received from the gateway and adding 7/8 (87.5%) of the time elapsed since that packet was received. If the SSSOE in the newly received packet is less than the client's conservative estimate by more than one second, then the client concludes that the NAT gateway has undergone a reboot or other loss of port mapping state, and the client MUST immediately renew all its active port mapping leases as described in Section 3.7 "Recreating Mappings On NAT Gateway Reboot".

3.7 Recreating Mappings On NAT Gateway Reboot

The NAT gateway MAY store mappings in persistent storage so when it is powered off or rebooted, it remembers the port mapping state of the network.

However, maintaining this state is not essential for correct

operation. When the NAT gateway powers on or clears its port mapping state as the result of a configuration change, it MUST reset the epoch time and re-announce its IP address as described in Section 3.2.1 "Announcing Address Changes". Reception of this packet where time has apparently gone backwards serves as a hint to clients on the network that they SHOULD immediately send renewal packets (to immediately recreate their mappings) instead of waiting until the originally scheduled time for those renewals. Clients who miss receiving those gateway announcement packets for any reason will still renew their mappings at the originally scheduled time and cause their mappings to be recreated; it will just take a little longer for these clients.

A mapping renewal packet is formatted identically to an original mapping request; from the point of view of the client it is a renewal of an existing mapping, but from the point of view of the freshly-rebooted NAT gateway it appears as a new mapping request.

This self-healing property of the protocol is very important.

The remarkable reliability of the Internet as a whole derives in large part from the fact that important state is held in the endpoints, not in the network itself [ETEAISD]. Power-cycling an Ethernet switch results only in a brief interruption in the flow of packets; established TCP connections through that switch are not broken, merely delayed for a few seconds. Indeed, an old Ethernet switch can even be replaced with a new one, and as long as the cables are transferred over reasonably quickly, after the upgrade all the TCP connections that were previously going though the old switch will be unbroken and now going through the new one. The same is true of IP routers, wireless base stations, etc. The one exception is NAT gateways. Because the port mapping state is required for the NAT gateway to know where to forward inbound packets, loss of that state breaks connectivity through the NAT gateway. By allowing clients to detect when this loss of NAT gateway state has occurred, and recreate it on demand, we turn hard state in the network into soft state, and allow it to be recovered automatically when needed.

Without this automatic recreation of soft state in the NAT gateway, reliable long-term networking would not be achieved. As mentioned above, the reliability of the Internet does not come from trying to build a perfect network in which errors never happen, but from accepting that in any sufficiently large system there will always be some component somewhere that's failing, and designing mechanisms

that can handle those failures and recover. To illustrate this point with an example, consider the following scenario: Imagine a network security camera that has a web interface and accepts incoming connections from web browser clients. Imagine this network security camera uses NAT-PMP or a similar protocol to set up an inbound port mapping in the NAT gateway so that it can receive incoming connections from clients the other side of the NAT gateway.
Now, this camera may well operate for weeks, months, or even years. During that time it's possible that the NAT gateway could experience a power failure or be rebooted. The user could upgrade the NAT gateway's firmware, or even replace the entire NAT gateway device with a newer model. The general point is that if the camera operates for a long enough period of time, some kind of disruption to the NAT gateway becomes inevitable. The question is not whether the NAT gateway will lose its port mappings, but when, and how often.
If the network camera and devices like it on the network can detect when the NAT gateway has lost its port mappings, and recreate them automatically, then these disruptions are self-correcting and invisible to the end user. If, on the other hand, the disruptions are not self-correcting, and after a NAT gateway reboot the user has to manually reset or reboot all the other devices on the network too, then these disruptions are *very* visible to the end user. This aspect of the design is what makes the difference between a protocol that keeps on working indefinitely over a time scale of months or years, and a protocol that works in brief testing, but in the real world is continually failing and requiring manual intervention to get it going again.

When a client renews its port mappings as the result of receiving a packet where the "Seconds since start of epoch" field (SSSOE) indicates that a reboot or similar loss of state has occurred, the client MUST first delay by a random amount of time selected with uniform random distribution in the range 0 to 5 seconds, and then send its first port mapping request. After that request is acknowledged by the gateway, the client may then send its second request, and so on, as rapidly as the gateway allows. The requests SHOULD be issued serially, one at a time; the client SHOULD NOT issue multiple requests simultaneously in parallel.

The discussion in this section focusses on recreating inbound port mappings after loss of NAT gateway state, because that is the more serious problem. Losing port mappings for outgoing connections destroys those currently active connections, but does not prevent clients from establishing new outgoing connections. In contrast, losing inbound port mappings not only destroys all existing inbound connections, but also prevents the reception of any new inbound connections until the port mapping is recreated. Accordingly, we consider recovery of inbound port mappings the more important priority. However, clients that want outgoing connections to survive

a NAT gateway reboot can also achieve that using NAT-PMP. After initiating an outbound TCP connection (which will cause the NAT

gateway to establish an implicit port mapping) the client should send the NAT gateway a port mapping request for the source port of its TCP connection, which will cause the NAT gateway to send a response giving the public port it allocated for that mapping. The client can then store this information, and use later to recreate the mapping if it determines that the NAT gateway has lost its mapping state.

3.8 NAT Gateways with NAT Function Disabled

Note that *only* devices currently acting in the role of NAT gateway should participate in NAT-PMP protocol exchanges with clients. A network device that is capable of NAT (and NAT-PMP), but is currently configured not to perform that function, (e.g. it is acting as a traditional IP router, forwarding packets without modifying them), MUST NOT respond to NAT-PMP requests from clients, or send spontaneous NAT-PMP address-change announcements.

In particular, a network device not currently acting in the role of NAT gateway should not even respond to NAT-PMP requests by returning an error code such as "2 - Not Authorized/Refused", since to do so is misleading to clients -- it suggests that NAT port mapping is necessary on this network for the client to successfully receive inbound connections, but is not available because the administrator has chosen to disable that functionality.

Clients should also be careful to avoid making unfounded assumptions, such as the assumption that if the client has an IPv4 address in one of the RFC 1918 private IP address ranges then that means NAT necessarily must be in use. Net 10/8 has enough addresses to build a private network with millions of hosts and thousands of interconnected subnets, all without any use of NAT. Many organizations have built such private networks that benefit from using standard TCP/IP technology, but by choice do not connect to the public Internet. The purpose of NAT-PMP is to mitigate some of the damage caused by NAT. It would be an ironic and unwanted side-effect of this protocol if it were to lead well-meaning but misguided developers to create products that refuse to work on a private network *unless* they can find a NAT gateway to talk to. Consequently, a client finding that NAT-PMP is not available on its

network should not give up, but should proceed on the assumption that the network may be a traditional routed IP network, with no address translation being used. This assumption may not always be true, but it is better than the alternative of falsely assuming the worst and not even trying to use normal (non-NAT) IP networking.

If a network device not currently acting in the role of NAT gateway receives UDP packets addressed to port 5351, it SHOULD respond immediately with an "ICMP Port Unreachable" message to tell the client that it needn't continue with timeouts and retransmissions, and it should assume that NAT-PMP is not available and not needed on this network.

4. UNSAF Considerations

The document "IAB Considerations for UNSAF Across NAT" [RFC 3424] covers a number of issues when working with NATs. RFC 3424 outlines some requirements for any document that attempts to work around problems associated with NATs. This section addresses those requirements.

4.1 Scope

This protocol addresses the needs of TCP and UDP transport peers that are separated from the public internet by exactly one NAT. Such peers must have access to some form of directory server for registering the public IP address and port at which they can be reached.

4.2 Transition Plan

Any client making use of this protocol SHOULD implement IPv6 support. If a client supports IPv6 and is running on a device with a global IPv6 address, that IPv6 address SHOULD be preferred to the IPv4 public address using this NAT mapping protocol. In case other clients do not have IPv6 connectivity, both the IPv4 and IPv6 addresses SHOULD be registered with whatever form of directory server is used. Preference SHOULD be given to IPv6 addresses when available. By implementing support for IPv6 and using this protocol for IPv4, vendors can ship products today that will work under both scenarios. As IPv6 is more widely deployed, clients of this protocol following these recommendations will transparently make use of IPv6.

4.3 Failure Cases

Aside from NATs that do not implement this protocol, there are a number of situations where this protocol may not work.

4.3.1 NAT Behind NAT

Some people's primary IP address, assigned by their ISP, may itself be a NAT address. In addition, some people may have a public IP address, but may then double NAT themselves, perhaps by choice or perhaps by accident. Although it might be possible in principle for one NAT gateway to recursively request a mapping from the next one, this document does not advocate that and does not try to prescribe how it would be done.

It would be a lot of work to implement nested NAT port mapping correctly, and there are a number of reasons why the end result might

not be as useful as we might hope. Consider the case of an ISP that offers each of its customers only a single NAT address. This ISP could instead have chosen to provide each customer with a single public IP address, or, if the ISP insists on running NAT, it could have chosen to allow each customer a reasonable number of addresses, enough for each customer device to have its own NAT address directly from the ISP. If instead this ISP chooses to allow each customer just one and only one NAT address, forcing said customer to run nested NAT in order to use more than one device, it seems unlikely that such an ISP would be so obliging as to provide a NAT service that supports NAT Port Mapping Protocol. Supposing that such an ISP did wish to offer its customers NAT service with NAT-PMP so as to give them the ability to receive inbound connections, this ISP could easily choose to allow each client to request a reasonable number of DHCP addresses from that gateway. Remember that Net 10/8 [RFC 1918] allows for over 16 million addresses, so NAT addresses are not in any way in short supply. A single NAT gateway with 16 million available addresses is likely to run out of packet forwarding capacity before it runs out of private addresses to hand out. In this way the ISP could offer single-level NAT with NAT-PMP, obviating the need to support nested NAT-PMP. In addition, an ISP that is motivated to provide their customers with unhindered access to the Internet by

allowing incoming as well as outgoing connections has better ways to offer this service. Such an ISP could offer its customers real public IP addresses instead of NAT addresses, or could even choose to offer its customers full IPv6 connectivity, where no mapping or translation is required at all.

4.3.2 NATs with Multiple Public IP Addresses

If a NAT maps private addresses to multiple public addresses, then it SHOULD pick one of those addresses as the one it will support for inbound connections, and for the purposes of this protocol it SHOULD act as if that address were its only address.

4.3.3 NATs and Routed Private Networks

In some cases, a large network may be subnetted. Some sites may install a NAT gateway and subnet the private network. Such subnetting breaks this protocol because the router address is not necessarily the address of the device performing NAT.

Addressing this problem is not a high priority. Any site with the resources to set up such a configuration should have the resources to add manual mappings or attain a range of globally unique addresses.

Not all NATs will support this protocol. In the case where a client is run behind a NAT that does not support this protocol, the software relying on the functionality of this protocol may be unusable.

4.3.4 Communication Between Hosts Behind the Same NAT

NAT gateways supporting NAT-PMP should also implement "hairpin translation". Hairpin translation means supporting communication between two local clients being served by the same NAT gateway.

Suppose device A is listening on private address and port 10.0.0.2:80 for incoming connections. Using NAT-PMP, device A has obtained a mapping to public address and port x.x.x.x:80, and has recorded this public address and port in a public directory of some kind. For example, it could have created a DNS SRV record containing this information, and recorded it, using DNS Dynamic Update [RFC 3007], in a publicly accessible DNS server. Suppose then that device B, behind

the same NAT gateway as device A, but unknowing or uncaring of this fact, retrieves device A's DNS SRV record and attempts to open a TCP connection to x.x.x.x:80. The outgoing packets addressed to this public Internet address will be sent to the NAT gateway for translation and forwarding. Having translated the source address and port number on the outgoing packet, the NAT gateway needs to be smart enough to recognize that the destination address is in fact itself, and then feed this packet back into its packet reception engine, to perform the destination port mapping lookup to translate and forward this packet to device A at address and port 10.0.0.2:80.

4.3.5 Non UDP/TCP Transport Traffic

Any communication over transport protocols other than TCP and UDP will not be served by this protocol. Examples are Generic Routing Encapsulation (GRE), Authentication Header (AH) and Encapsulating Security Payload (ESP).

4.4 Long Term Solution

As IPv6 is deployed, clients of this protocol supporting IPv6 will be able to bypass this protocol and the NAT when communicating with other IPv6 devices. In order to ensure this transition, any client implementing this protocol SHOULD also implement IPv6 and use this solution only when IPv6 is not available to both peers.

4.5 Existing Deployed NATs

Existing deployed NATs will not support this protocol. This protocol will only work with NATs that are upgraded to support it.

5. Security Considerations

As discussed in section 3.2 "Determining the Public Address", only clients on the private side of the NAT may create port mappings, and

only on behalf of themselves. By using IP address spoofing, it's possible for one client to delete the port mappings of another client. It's also possible for one client to create port mappings on behalf of another client. The best way to deal with this vulnerability is to use IPSec [RFC 2401].

Since allowing incoming connections is often a policy decision, any NAT gateway implementing this protocol SHOULD have an administrative mechanism to disable it.

Some people view the property that NATs block inbound connections as a security benefit which is undermined by this protocol. The authors of this document have a different point of view. In the days before NAT, all hosts had unique public IP addresses, and had unhindered ability to communicate with any other host on the Internet. When NAT came along it broke this unhindered connectivity, relegating many hosts to second-class status, unable to receive inbound connections. This protocol goes some way to undo some of that damage. The purpose of a NAT gateway should be to allow several hosts to share a single address, not to simultaneously impede those host's ability to communicate freely. Security is most properly provided by end-to-end cryptographic security, and/or by explicit firewall functionality, as appropriate. Blocking of certain connections should occur only as a result of explicit and intentional firewall policy, not as an accidental side-effect of some other technology.

6. IANA Considerations

No IANA services are required by this document.

7. Acknowledgments

The concepts described in this document have been explored, developed and implemented with help from Bob Bradley, Josh Graessley, Rob Newberry, Roger Pantos, John Saxton, and James Woodyatt.

8. Deployment History

NAT-PMP client software first became available to the public through Apple's Darwin Open Source code in August 2004. NAT-PMP implementations began shipping to end users in large volumes (i.e. millions) with the launch of Mac OS X 10.4 Tiger and Bonjour for Windows 1.0 in April 2005.

The NAT-PMP client in Mac OS X 10.4 Tiger and Bonjour for Windows exists as part of the mDNSResponder system service. When a client advertises a service using Wide Area Bonjour [DNS-SD], and the machine is behind a NAT-PMP-capable NAT gateway, then if the machine is so configured, the mDNSResponder system service automatically uses NAT-PMP to set up an inbound port mapping, and then records the public IP address and port in the global DNS. Existing client software using the existing Bonjour programming APIs [Bonjour] gets this functionality automatically. The logic is that if client software publishes its information into the global DNS via Wide Area Bonjour service advertising, then it's reasonable to infer an expectation that this information should be usable by the peers retrieving it. Generally speaking, recording a private IP address like 10.0.0.2 in the public DNS is completely pointless because that address is not reachable from clients on the other side of the NAT gateway. In the case of a home user with a single computer directly connected to their Cable or DSL modem, with a single global IPv4 address and no NAT gateway (a surprisingly common configuration), publishing that IP address into the global DNS is useful because that IP address is reachable. In contrast, a home user using a NAT gateway to share a single global IPv4 address between several computers loses this ability to receive inbound connections easily. This breaks many peer-to-peer collaborative applications, like the multi-user text editor SubEthaEdit [SEE]. Automatically creating the necessary inbound port mappings helps remedy this unintended side-effect of NAT.

The server side of the NAT-PMP protocol is implemented in Apple's "AirPort Extreme" and "AirPort Express" wireless base stations.

9. Copyright Notice

WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

10. Normative References

[RFC 1918] Y. Rekhter et.al., "Address Allocation for Private Internets", RFC 1918, February 1996.

[RFC 2119] RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels

11. Informative References

[Bonjour] Apple "Bonjour" <http://developer.apple.com/bonjour/>

[ETEAISD] J. Saltzer, D. Reed and D. Clark: "End-to-end arguments in system design", ACM Trans. Comp. Sys., 2(4):277-88, Nov. 1984

[DNS-SD] Cheshire, S., and M. Krochmal, "DNS-Based Service Discovery", Internet-Draft (work in progress), draft-cheshire-dnsext-dns-sd-04.txt, August 2006.

[mDNS] Cheshire, S., and M. Krochmal, "Multicast DNS", Internet-Draft (work in progress), draft-cheshire-dnsext-multicastdns-06.txt, August 2006.

[RFC 2131] R. Droms, "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC 2401] Atkinson, R. and S. Kent, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC 2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

[RFC 3007] Wellington, B., "Simple Secure Domain Name System (DNS) Dynamic Update", RFC 3007, November 2000.

[SEE] <http://www.codingmonkeys.de/subethaedit/>

[RFC 3022] RFC 3022 - Network Address Translator

[RFC 3424] RFC 3424 - IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation

12. Authors' Addresses

Stuart Cheshire
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 3207
EMail: rfc [at] stuartcheshire [dot] org

Marc Krochmal
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA

Phone: +1 408 974 4368
EMail: marc [at] apple [dot] com

Kiren Sekar
Sharpcast, Inc.
250 Cambridge Ave, Suite 101
Palo Alto
California 94306

USA

Phone: +1 650 323 1960
EMail: ksekar [at] sharpcast [dot] com