

## RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

---

*Source:*

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2007-03/msg00460.1>

---

- *From:* Anton Bassov <[AntonBassov@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:AntonBassov@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 19 Mar 2007 20:40:08 -0700
- 

Our windows system has at least TWO network adapter. So, I cannot sure that the destination NIC of IPv6 packet is the same as the destination NIC of my encapped IPv4 packet.

And, in such case, you just cannot say that you have IPv4-to-IPv6 address mapping.....

Before you proceed to the solution described by Steve, I would advise you to think

about how you are going to map IPv6 addresses to IPv4 ones – even if you choose the approach that Steve has described, you still need to map addresses. Therefore, no matter how you look at it, you still need do deal with ARP, no matter how much you want to avoid it.

In general, you should clearly realize that emulating non-existent IPv6 stack is quite challengeing task in itself – even for someone who is experienced with NDIS. For the time being you don't seem to see a difference between being TCPIP client and indicating packets to it. Therefore, you will have to a lot of work and a lot of learning –if you don't want to do it because it all seems too complex for you, then you should not even start this project

Anton Bassov

"Seong Moon" wrote:

Hi Anton!

First, thanks for your sincere comment !

Once you have IPv6-to-IPv4 destination address mapping, it somehow implies that the actual destination NIC is the same for both IPv6 and IPv4 addresses

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

(actual source is the same for understandable reasons – it is just sender's NIC). Therefore, no matter if you use IPv4 or IPv6, the \*physical\* route is still the same, and, hence, source and destination MAC addresses are the same for both IP versions.

That's not the our case.

Our windows system has at least TWO network adapter. So, I cannot sure that the destination NIC of IPv6 packet is the same as the destination NIC of my encapped IPv4 packet.

Actually, how are you going to build IPv6-to-IPv4 address translation table without knowing MAC addresses ???

That is the reason why I posted the question a first. I was wondering whether the tcpip stack can forward the encapped packet or not. Of course, I was assuming that tcpip stack can rebuild the L2 header for the encapped IPv4 packet which is indicated to upper tcpip stack on my MiniportSendpackets().

"Anton Bassov" wrote:

On one hand, you said the following:

My IM driver has a mapping table which consists of

- > o. destination IPv6 address
- > o. local IPv4 address
- > o. remote IPv4 address

\*\*\*\*\*

On another hand, you said the following:

The original MAC header is INVALID because my IM driver will assign a new destination IPv4 address and the original MAC header was built for original IPv6 destination IPv6 address.

Don't you see any logical contradiction between these 2 statements (I assume you understand what MAC address is) ???

Once you have IPv6-to-IPv4 destination address mapping, it somehow implies that the actual destination NIC is the same for both IPv6 and IPv4 addresses (actual source is the same for understandable reasons – it is just sender's

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM  
NIC). Therefore, no matter if you use IPv4 or IPv6, the \*physical\* route is  
still the same, and, hence,  
source and destination MAC addresses are the same for both IP versions.

Actually, how are you going to build IPv6-to-IPv4 address translation table  
without  
knowing MAC addresses ??? Has it ever occurred to you to think that MAC  
address is just a missing link between these two???

Anton Bassov

"Seong Moon" wrote:

In the sending procedure,  
The original MAC header is INVALID because my IM  
driver will assign a new  
destination IPv4 address and the original MAC header was  
built for original  
IPv6 destination IPv6 address.

After prepending IPv4 header and UDP header to the original  
IPv6 packet, the  
original MAC header is useless. That is the reason why I  
indicate the  
encapped IPv4 packet up to the TCP/IP layer.

So, I can't agree with that you mentioned "forward the packet  
to miniport  
with NdisSend()". Do I misunderstand ?

regards seong

"Anton Bassov" wrote:

As I told you already, you should NOT  
indicate outgoing packet to TCPIP.  
Instead, you should just prepend UDP and  
IPv4 IP headers to IPv6 packet  
without removing MAC header, then  
prepend MAC header to it (if we assume that  
TCPIP has chosen the appropriate adapter,  
then source and destination MAC  
addresses in the original MAC header are  
still valid), and then forward the  
packet to miniport with NdisSend().  
This is how you should deal with sends

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

When it comes to receives, you should detect IPv4 packet of interest, and remove its MAC, IPv4 and UDP headers. As a result, you will get exactly the same IPv6 packet (i.e. with its MAC, IPv4 and UDP headers) that was passed to your NDIS IM on the sender machine, so that you just indicate this packet to NDIS

Anton Bassov

"Seong Moon" wrote:

Hi Anton!

I'm the one who posted the question about IPv6 over IPv4 UDP tunnel driver.

I'm now thinking the IM driver for my objective.

In the process of packet transmission, I have a following scenario.

1. IPv6 application sends a IPv6 packet.
2. The packet can be captured on my IM driver. The captured packet will be as following

---

| L2 header| IPv6 header |  
Payload |

---

3. My IM driver has a mapping table which consists of
  - o. destination IPv6 address
  - o. local IPv4 address
  - o. remote IPv4 address
  - o. local UDP Port number
  - o. remote UDP Port numberthe above information should set correctly via the other control

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

application.

4. If the captured IPv6 packet on my IM driver has a destination IPv6 address which has already configured on the above mapping table, the packet should be processed as follows :

- o. The IPv4 header and UDP header are prepended on the original IPv6 packet.
- o. Of course, L2 header(e.g. MAC header) should be ignored.
- o. So, the final packet will be the following format.

---

| IPv4 header | UDP header |  
IPv6 header | Payload |

---

5. Now, I can indicate the encapsulated packet the the upper TCP/IP layer. As result, I'd like to have the IP layer forward the IPv4 packet correctly.

6. As you mentioned, the MAC header should be processed correctly. I'm not sure how I can process the MAC header. Any idea ?

Until here is my idea about the transmission of the tunneled packet.

I'll appreciate any comment.

regards seong

"Anton Bassov" wrote:

Can  
the

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

TCP/IP  
driver  
forward  
the  
packet  
correctly  
?

Forward to  
whom????  
MiniportSendPackets()  
deals only  
with  
outgoing  
packets,  
which  
means that  
packet's  
source IP  
and MAC  
addresses  
are local  
and  
destination  
ones are  
remote, i.e.  
the packet is  
outgoing  
one and  
TCP/IP  
already  
made its  
routing  
decision to  
send a  
packet via  
either  
underlying  
adapter (if  
you speak  
about filter),  
or via your  
virtual  
miniport (if  
you  
speak about  
MUX). If  
you want to  
change the  
adapter the

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

packet gets  
sent  
through,  
you have to  
call  
NdisSend()  
and, specify  
NDIS\_HANDLE  
that  
corresponds  
to that  
adapter (  
apparently,  
it is a good  
idea to  
replace  
MAC  
addresses in  
a packet's  
ARP header  
as well),  
rather than  
indicating it  
to TCPIP as  
incoming  
one.  
Apparently,  
TCPIP is  
going to  
silently  
discard  
your packet  
if you  
indicate it,  
and that's  
it.....

If you want  
TCPIP to  
treat you  
packet as  
incoming  
one, you  
have to  
modify  
both  
physical  
and logical  
transport  
addresses in  
the packet's

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

headers.  
However, if  
TCPIP  
takes it for  
incoming  
one, why  
should it  
send it to  
any remote  
address(unless  
logical IP  
address in  
the packet's  
header is  
remote and  
the  
system is  
configured  
as a  
router)???

In other  
words, no  
matter what  
you do, you  
have to  
modify  
packet's  
headers  
in such way  
that it  
conforms to  
the logic of  
the bound  
protocol...

Another  
option (at  
least in  
MUX)  
could be  
taking the  
whole  
packet  
simply as  
data, and  
resubmitting  
it to TCPIP  
via its TDI  
interface,  
i.e. to act

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

simply  
as TCP/IP's  
client

If you tell  
us a bit  
more about  
your  
objectives,  
probably,  
we will be  
able  
to give you  
the most  
appropriate  
advice....

Anton  
Bassov

"Seong  
Moon"  
wrote:

Hi  
!  
There  
!

I'm  
now  
designing  
Intermediate  
Driver  
for  
my  
project.

I'd  
like  
to  
transfer  
sending  
packet  
to  
the  
upper

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

TCP/IP  
protocol  
layer  
on  
my  
MiniportSendPackets  
function  
as  
if  
the  
packet  
is  
received  
from  
the  
NIC.  
And  
I'd  
like  
to  
have  
IP  
protocol  
layer  
forward  
the  
transferred  
packet  
to  
the  
adaquate  
adaptor.

Is  
it  
possible  
?

That  
is,  
Can  
I  
call  
NdisMIndicateReceivePacket()  
in  
my  
MiniportSendPacket()  
?  
If  
that  
possible,

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

RE: Transfer a sending packet to upper TCP/IP protocol layer in IM

Can  
the  
TCP/IP  
driver  
forward  
the  
packet  
correctly  
?

Thanks  
in  
advance.  
regards  
seong