Click Frequently Asked Questions

Johan Bergs — Jeremy Van den Eynde

Contents

1 General

Which sources apart from this FAQ and the Click tutorial are good to learn more about Click?

The Click Modular Router homepage `http://www.read.cs.ucla.edu/click/`.

I have asked everyone and no one can help me with my Click problems. Can I now ask for help on the Click malinglist?

NO. Mail to johan.bergs@uantwerpen.be and jeremy.vandeneynde@uantwerpen.be, they will help you. (If everyone mails simple questions to the mailinglist then the high quality of it will drop, with less and worse Click releases as a consequence.)

I run Click on my 64bit CPU and configure fails saying that it does not know the machine 'x86_64-unknown'?

Replace 'config.sub' by /usr/share/automake/config.sub and rerun configure. This version is probably more up to date and will support your architecture.

On my 64bit CPU I get a compilation error in aqm/red.cc about a cast from void* to int.

You are running Click-1.4.3! It was not yet ready for 64 bit systems, where this type of casts is not permitted. Replace the line with a correct cast:

```
|   switch ((intptr* t)vparam) {
```

Click does not compile on (Open)Solaris

Click can be compiled on a Solaris system, with some tweaks to the makefile. But Fenix is very slow and the platform does not correspond to the reference platform. We strongly discourage you to compile on Fenix.

2 Scripts

ToDump writes to a PCAP dumpfile to analyse things in Wireshark, but nothing reasonable appears in the dumps?

The timestamp is not set on your packets. Using the SetTimeStamp element in front of the ToDump element will solve this.

ToDump logs only a fraction of what passes through it?

Do not log with multiple ToDumps to the same file, as only one of them will write out passed packets properly. Use separate files for each ToDump element.

I want to dump IP packets that are not encapsulated in Ethernet headers. How can I do that without encapsulating with fake Ethernet headers first?

Take a look at the ENCAP keyword for the ToDump element.

I can't use that EtherSwitch or ListenEtherSwitch element, Click does not find it?

You should enable the etherswitch package while configuring Click, so: ./configure —disable-linuxmodule —enable-local —enable-etherswitch

Is there syntax highlighting for click scripts in my favourite editor?

Most probably not directly, but syntax highlighting for C++ works well and makes reading click scripts sometimes a little bit easier. *Hint:Kwrite/Kate allows you to expand and collapse compound elements, which can help reading large scripts.*

So, you seem to need to run packets through something like CheckIPHeader in order to fill out the packet header annotations, correct? Do these annotations persist across elements that use "uniquify", things like Tee, etc.?

Yes.

Are transport-layer annotations only filled out once you run through something like IPClassifier?

The CheckIPHeader annotation sets both the start-network-header annotation and the end-network-header==start-transport-header annotation.

Why am I getting messages that no route is found for 0.0.0.0?

If packets do not contain an IP destination annotation, the routing table will try to route those packets based on an empty annotation, which equals trying to find a route for the address 0.0.0.0. So setting that annotation is crucial.

I want to broadcast packets. Should I add a route for 255.255.255.255?

No! There are multiple reasons. Broadcast packets should not be routed at all, as they are link local. You will also know the L2 address, most probably Ethernet broadcast. Sending these packets to an ARPQuerier is not a good idea. Also, with routers with multiple interfaces, where are you going to send the packet to?

I get messages that no route is found for 255.255.255.255?

Think about why you get these messages and take a good look at the previous question.

3 API

Where can I find a browsable version of the Click code?

The doxygen documentation is located at `http://www.read.cs.ucla.edu/click/doxygen/`

How do I use a hashmap without compilation errors?

Do not forget to include bighashmap.cc in your sourcefile with the right arguments. It's best to do it at the bottom, so you get something like:

```
// macro magic to use bighashmap
#include <click/bighashmap.cc>
#if EXPLICIT_TEMPLATE_INSTANCES
template class HashMap <Packet*, IPAddress*> ;
#endif
CLICK_ENDDECLS
EXPORT_ELEMENT(MyElement)
```

Am I right in thinking that variables like "network_length()" represent the length from the beginning of the network header to the end of the packet?

Yes!

Am I right in assuming there will never be any layering information (e.g., ethernet footer) after the data?

In practice yes, although this depends on your network layer.

Does changing the annotation variables (e.g., ip_header()->ip_len) actually modify the packet itself?

Nope!

How can I use multiple structs to construct a packet?

This is actually a matter of doing pointer arithmetics. Yes it's C, yes it's dangerous but yes you need it.
Say you have a packet format for the first part, defined in the struct FirstPart. The next part of the packet is defined in the struct NextPart.
When you want to create a packet with those headers encapsulated in Ethernet, IP and UDP, you will do almost the same as with a packet consisting of one struct. The only difference if the packet size and of course the way to access the data.

```
int tailroom = 0;
int packetsize = sizeof(FirstPart)+sizeof(SecondPart);
int headroom = sizeof(click_eth)+sizeof(clickip)+sizeof(
    clickudp);
WritablePacket* packet = Packet::make(headroom, 0,packetsize,
    tailroom);
if (packet == 0 )return click_chatter( "cannot make packet!");
memset(packet->data(), 0, packet->length());
FirstPart* format = (FirstPart*) packet->data();
format->something = somevalue;
NextPart* nextformat = (NextPart*) (packet->data()+sizeof(
    FirstPart));
nextformat->somethingelse = someothervalue;
```

Note that a different way to achieve this is doing:

```
struct NextPart{
SomethingElseType somethingelse;
}
struct FirstPart{
SomethingType something;
// ...
struct NextPart;
}
```

This also works but it limits the flexibility, with the first solution you can easily manage dynamic packet sizes.

4 New elements

My new element is not found, it does not appear in elements.conf. What's going on?

Formatting: check if no space ended up before EXPORT_ELEMENT. If that's the case Click ignores that macro so your element is not added.

I have made a directory below elements to group a number of elements, but these elements are not found by Click.

Look in configure.in for ELEMENTS_ARG_ENABLE, these are the statements for the inclusion of element directories. Now add a line of the form:

```
ELEMENTS\_ARG\_ENABLE(dirname, {[}comments{]}, NO)
```

The last option is either YES or No and determines whether the elements in your directory should be included by default when executing configure.
When you write YES here: execute configure again, if you do not want to include elements in the new directory then add the option —disable-dirname to your configure statement.
When you write NO here: execute configure again, but only when adding —enable-dirname these elements will be included. Do not forget to execute `make elemlist` again when adding more elements to the directory.

I have moved an element or changed my directory structure and all of a sudden I get errors that Click can not find the required files for my element?

In that case you need to run a make clean and execute configure again, of course with the right options enabled or disabled. On top in 'config.log' you can find which parameters you used last time, if you do not remember them anymore.

5 Compilation and link errors

Click does not want to compile any more, the compilation error is located in the 'beetlemonkey' function?

Your compiler output will look similar to this:

```
elements.cc: In function 'Click::Element* beetlemonkey(unsigned
     int)':
elements.cc:284: error: ISO C++ forbids declaration of 'type
    name' with no type
elements.cc:284: error: uninitialized const in 'new' of 'const
    int'
elements.cc:284: error: cannot convert 'const int*' to 'Click::
    Element*' in return
elements.cc:285: error: cannot convert 'char**' to 'Click::
    Element*' in return
elements.cc:286: error: syntax error before '(' token
elements.cc:287: error: ISO C++ forbids declaration of 'type
    name' with no type
elements.cc:287: error: uninitialized const in 'new' of 'const
    int'
elements.cc:287: error: cannot convert 'const int*' to 'Click::
    Element*' in return
elements.cc: In function 'void click_export_elements()':
elements.cc:582: error: syntax error before '(' token
elements.cc:586: error: 'class_name' undeclared (first use this
    function)
elements.cc:586: error: (Each undeclared identifier is reported
    only once for each function it appears in.)
elements.cc:586: error: syntax error before '::' token
elements.cc:588: error: syntax error before '(' token
```

```
elements.cc: In function 'void click_unexport_elements()':
elements.cc:906: error: syntax error before '(' token
elements.cc:908: error: syntax error before '::' token
elements.cc:909: error: syntax error before '(' token
gmake[1]: *** [elements.o] Error 1
```

In the header file of one of your elements the method 'class_name' is not defined on 1 line. This will work:

```
const char* class_name() const { return "OLSRNeighborInfoBase";
    }
```

The following will NOT work:

```
const char* class_name() const
{
return "OLSRNeighborInfoBase";
}
```

To solve this, you have to find out which element contains the errors, which is possible by analyzing the compiler output to elements.cc. You will see that something is wrong there. The element that causes the problems can be located quickly. Make sure 'class_name' is fixed and rerun `make elemlist all` to recompile. After this those compiler errors should be gone.
Hint:Take care with editors that format your source code, they might not leave 'class_name' on 1 line.

My error looks like the previous one but I certainly have everything on 1 line?

Check if no extra spaces appear after the () in EXPORT_ELEMENT. That "macro" is very sensitive.

My error looks like the previous one but EXPORT_ELEMENT is fine too. Click complains that certain elements do not exist?

Errors like this one:

```
CXX elements.cc
elements.cc: In function
Click::Element* beetlemonkey(uintptr_t)':
elements.cc:349: error:
Print' has not been declared
make[1]: *** [elements.o] Error 1
```

or

```
elements.cc: In function 'Element* beetlemonkey(uintptr_t)':
elements.cc:259: error: expected type-specifier before '
    MyNullPush'
elements.cc:259: error: cannot convert 'int*' to 'Element*' in
    return
elements.cc:259: error: expected ';' before 'MyNullPush'
elements.cc:259: error: 'MyNullPush' was not declared in this
    scope
make[1]: *** [elements.o] Error 1
```

arise when an includeguard is not copied well from the original code. Most probably the same name is used in 2 elements. So check your #ifndef and #define macros.

I get linkerrors about a vtable in one of my elements?

Errors like this one:

```
../elements/pdhcp/pdhcp_discover_generator.cc:14: undefined
    reference to
vtable for PDHCPDiscoverGenerator'
../elements/pdhcp/pdhcpdiscovergenerator.cc:14: undefined
    reference to
vtable for PDHCPDiscoverGenerator'
collect2: ld returned 1 exit status
```

are caused by not declaring or defining your element constructor and destructor.

gcc says the type ErrorHandler is not defined?

```
../elements/mobileiphelpers/dynamicetherencap.cc:47: error:
    invalid use of undefined type struct 'ErrorHandler'
```

Add this include at the top of your code:

```
#include <click/error.hh>
```

I get syntax error inside Click code, to be exact the includes?

Check whether you included click/config.h before everything else. If you did not, some macros are undefined with strange syntax errors as a consequence.

I get strange constructor deprecation errors?

If you get errors like this:

```
../elements/local/simplepullelement.cc: In constructor '
    SimplePullElement::SimplePullElement()':
../elements/local/simplepullelement.cc:8: warning: '_basector
    _' is deprecated (declared at ../include/click/element.hh
    :201)
```

then you are using old Click-1.4.3 element syntax. You should use the Click-1.6 syntax as described in the slides.

gcc starts complaining about pcap libraries during linking?

You are using code that is compiled on a different machine, and the object files refer to libraries on different locations. A make clean should solve this.

6 Endianness - htonl/htonl - htons/ntohs - uintX_t

Some fields in my packets are completely wrong, they seem reversed?

A network uses *big endian*-byte order while e.g. the systems in the pc labs use *little endian* as they run the intel architecture. You can use the following functions to convert integers in network-byte order:

```
unsigned short htons(unsigned short hostshort); // 16 bit
unsigned long htonl(unsigned long hostlong); //32- bit
```

To convert from network byte order to system byte order you can use:

```
unsigned short ntohs(unsigned short networkshort); // 16 bit
unsigned long ntohl(unsigned long networklong); //32- bit
```

More information on this subject is available at `http://en.wikipedia.org/wiki/Htons#Endianness_in_networking`.

What about uint_64?

Try to avoid using that, if you need to then you should contact Bart and Johan.

There is no such thing as an uint4_t, and I need this for two adjacent fields?

Start from two uint8_t fields and join those with bit-shifting and binary operators. Say you need the bit sequence for 01100001, then you could do this:

```
uint8_t a = 6; // (00000110)
uint8_t b = 1; // (00000001)
uint8_t result = (a << 4) & b;
```

I convert a number with htonl/htons and then it becomes 0?

Are you sure the datatype of the field you write in is not uint8? With uint16 you use htons and for uint32 it's htonl. (You should avoid using uint64 due to possible alignments issues.)

Is this related: I can't get my UDP checksum correct, though everything is set correctly?

Are you sure you used htons to set the UDP header length? If not the checksum can never be set correctly and things will always fail, even if the checksum is calculated with SetUDPChecksum.

Can't we avoid this endianness problem with uint8_t (8bits), uint16_t (16bits) etc?

Those are data types for unsigned integers with x bits. uint8_t = 1 byte, so endianness does not have any impact here. For all larger datatypes, so datatypes consisting of multiple bytes, you must take endianness into account. In the end, this is about how data is stored in memory, not about how data should be interpreted.

7 Data structure alignment

sizeof(struct X) returns the wrong size?

Because of the way data is structured within the memory of a computer, the compiler wants to align this data properly. A CPU operates with word sized chunks, which are 4 byte long on a 32 bit processor and 8 byte long on a 64 bit

processor. Therefore, when you define a struct, the compiler will add padding bytes to any data structure that does not align properly with these word sized chunks. Take following code for example:

```
struct X {
uint8_t a;
uint8_t b;
uint16_t c;
};
```

On a 32 bit system sizeof (struct X) would return 4. However, on a 64 bit system, this would return 8, as 8 bytes is the smallest multiple of the word length (8) that can represents 4 bytes. If you want to be able the actual size of a struct instead of the padded one, the solution is to define your struct as follows:

```
struct __attribute__((__packed__)) X {
uint8_t a;
uint8_t b;
uint16_t c;
};
```

This statement will instruct your compiler to only make the struct as large as it needs to be to hold all data (actually, on a per-byte basis), ignoring alignment. Now, sizeof (struct X) will always return 4.

8 Segfaults and other runtime errors

My code crashes, how do I start debugging?

Take a look at valgrind, ddd and gdb. If you need more information about these tools just post a question or ask Bart and Johan.

My elements configures without errors and still I get an unspecified error?

Most probably you get something like: While configuring 'aninstance :: SomeElement': unspecified error Check if your configure method returns 0 at its end, it has to return an integer you know.

When running Click, it crashes on timer initialization code?

Are you sure you are initializing your timer in the initialize method of your element (as in the examples)? (Initializing your timers in the configure method of an element is a bad idea. Then the timer will look for its router object, which is not completely configured because it is configuring its elements with their configure method. To avoid this dependency circle you should initialize your timer in the elements initialize method and everything will work properly.)

My element duplicates a packet (it is sent on multiple output ports or is used multiple times) and a little bit later I get segmentation faults?

If you duplicate packets you have to clone them too. Otherwise the memory assigned to the packet is freed multiple times, causing segfaults.

My code is written using every possible standard, but still it crashes when generating the first packet?

In which method are you generating this packet? You should not generate any packets in the configure or initialize method. At that moment, the router is not yet initialized completely and other elements may still have to be initialized. This means things can and will crash.

9 Assignment and RFCs

I have seen that the KULeuven/... has put a protocol implementation online. Can I use it? Or base my code on it?

You searched very well, but that's not the purpose of the project. By the way, that implementation is complete, it can do more than the assignments requires. Making everything compatible with the scripts is a lot of work... We know a lot of implementations and we are actively looking for new ones, as with every project copying equals fraud and as a consequence reduces your points to 0.

I changed an address in an IP header and now my UDP checksum is incorrect?

The UDP RFC states:

> Checksum is the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets. The pseudo header conceptually prefixed to the UDP header contains the source address, the destination address, the protocol, and the UDP length. This information gives protection against misrouted datagrams. This checksum procedure is the same as is used in TCP.

(So far for the theoretical ISO/OSI layers) So add a SetUDPChecksum element after changing IP addresses.

10 Hints for the project

A number of things might be useful to know:

- ask questions about Click and the RFC as soon as possible
- it's not good if you are stuck too long tell us about problems as soon as possible
- also if there are tensions in your project group use the Click code as much as possible
- almost never do you have to change existing elements, mail us if you think you have to

- for the final evaluation everything has to compile, run without crashing and work conform the RFC, if software does not compile or run we can't give you points!
- if you prepare the final tar.gz, test it on a different machine before testing, one header file is easy to forget
- etherswitch is already included in Click, don't send that element to us
- make sure you can work with handlers and write down commonly used handlers in a file so you don't have to retype them again and again
- avoid hardcoded constants like 0x22 throughout your code, define it once (with a macro or a constant) to make things much more clear
- use Clicks headers for IP, UDP, ...

To highlight this one again: use existing Click code where you can, even if the task is very small. Nothing is easier than pushing out a packet and having the small task done by another element. You are sure that it works and don't need to wonder whether e.g. you first need to set a checksum to 0 to calculate it. So use existing code to:

- encapsulate a packet (dynamic destination? control the Encap element with handlers!)
- update checksums (UDP, IP, ...) or set them after packet creation
- strip some bytes, at the front or at the back of the packet
- set headers
- modify special headers, then use Click's structs instead of yours

Hint: tunnelling is possible without almost any new code.