

[RFC/PATCH] network configs: disconnect network options from drivers

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-03/9619.html>

From: Randy Dunlap (*randy.dunlap_at_verizon.net*)

Date: 03/31/05

Date: Wed, 30 Mar 2005 23:47:09 -0800

To: ioe-lkml@axxeo.de, matthew@wil.cx

RFC: This is a work-in-progress (WIP), not yet completed.

A few people dislike that the Networking Options menu is inside the Device Drivers/Networking menu. This patch moves the Networking Options menu to immediately before the Device Drivers menu, renames it to "Networking options and protocols", & moves most protocols to more logical places (IMHOOC).

The reasons that it is still WIP are:

- I'd like to see all of the sub-menus done in the same style;
- IrDA & Bluetooth subsystems include protocols & drivers, yet they are displayed under Networking protocols. I don't see much good reason to split them up. (See, this is an example of why the Networking Options and Network Drivers were close together....)
- some Networking options need to be qualified with CONFIG_NET

Any comments?

Signed-off-by: Randy Dunlap <rddunlap@osdl.org>

drivers/Kconfig | 4

drivers/net/Kconfig | 5

net/Kconfig | 396 ++++++

3 files changed, 208 insertions(+), 197 deletions(-)

diff -Naurp -X /home/rddunlap/doc/dontdiff-osdl linux-2612-rc1-pv/drivers/Kconfig

linux-2612-rc1-netconfig/drivers/Kconfig

--- linux-2612-rc1-pv/drivers/Kconfig 2005-03-01 23:38:26.000000000 -0800

+++ linux-2612-rc1-netconfig/drivers/Kconfig 2005-03-30 22:19:48.231418331 -0800

@@ -1,5 +1,7 @@

drivers/Kconfig

+source "net/Kconfig"

+

Linux-Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

```
menu "Device Drivers"

source "drivers/base/Kconfig"
@@ -28,7 +30,7 @@ source "drivers/message/i2o/Kconfig"

source "drivers/macintosh/Kconfig"

-source "net/Kconfig"
+source "drivers/net/Kconfig"

source "drivers/isdn/Kconfig"

diff -Naurp -X /home/rddunlap/doc/dontdiff-osdl linux-2612-rc1-pv/drivers/net/Kconfig
linux-2612-rc1-netconfig/drivers/net/Kconfig
--- linux-2612-rc1-pv/drivers/net/Kconfig 2005-03-30 22:03:48.861547540 -0800
+++ linux-2612-rc1-netconfig/drivers/net/Kconfig 2005-03-30 22:59:00.441523697 -0800
@@ -1,8 +1,9 @@
-
#
# Network device configuration
#

+menu "Network device support"
+
config NETDEVICES
    depends on NET
    bool "Network device support"
@@ -2556,3 +2557,5 @@ config NETCONSOLE
    If you want to log kernel messages over the network, enable this.
    See <file:Documentation/networking/netconsole.txt> for details.

+endmenu
+
diff -Naurp -X /home/rddunlap/doc/dontdiff-osdl linux-2612-rc1-pv/net/Kconfig
linux-2612-rc1-netconfig/net/Kconfig
--- linux-2612-rc1-pv/net/Kconfig 2005-03-30 22:03:57.503790207 -0800
+++ linux-2612-rc1-netconfig/net/Kconfig 2005-03-30 23:09:11.686664598 -0800
@@ -2,15 +2,18 @@
# Network configuration
#

-mmenu "Networking support"
+menu "Networking options and protocols"

config NET
    bool "Networking support"
+ default y
---help---
    Unless you really know what you are doing, you should say Y here.
    The reason is that some programs need kernel networking support even
    when running on a stand-alone machine that isn't connected to any
```

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

- other computer. If you are upgrading from an older kernel, you
- + other computer.
- +
+ If you are upgrading from an older kernel, you
should consider updating your networking tools too because changes
in the kernel and the tools often go hand in hand. The tools are
contained in the package net–tools, the location and version number
@@ –20,9 +23,6 @@ config NET
recommended to read the NET–HOWTO, available from
<<http://www.tldp.org/docs.html#howto>>.

- menu "Networking options"
- depends on NET
- config PACKET
tristate "Packet socket"
---help---
- @@ –81,6 +81,25 @@ config NET_KEY

Say Y unless you know what you are doing.

- +config NETPOLL
- + def_bool NETCONSOLE
- +
+config NETPOLL_RX
- + bool "Netpoll support for trapping incoming packets"
- + default n
- + depends on NETPOLL
- +
+config NETPOLL_TRAP
- + bool "Netpoll traffic trapping"
- + default n
- + depends on NETPOLL
- +
+config NET_POLL_CONTROLLER
- + def_bool NETPOLL
- +
+menu "Networking protocols"
- + depends on NET
- +
config INET
bool "TCP/IP networking"
---help---
- @@ –127,100 +146,6 @@ config IPV6

source "net/ipv6/Kconfig"

- menuconfig NETFILTER
- bool "Network packet filtering (replaces ipchains)"
- ---help---
- Netfilter is a framework for filtering and mangling network packets

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

- that pass through your Linux box.
-
- The most common use of packet filtering is to run your Linux box as
 - a firewall protecting a local network from the Internet. The type of
 - firewall provided by this kernel support is called a "packet
 - filter", which means that it can reject individual network packets
 - based on type, source, destination etc. The other kind of firewall,
 - a "proxy–based" one, is more secure but more intrusive and more
 - bothersome to set up; it inspects the network traffic much more
 - closely, modifies it and has knowledge about the higher level
 - protocols, which a packet filter lacks. Moreover, proxy–based
 - firewalls often require changes to the programs running on the local
 - clients. Proxy–based firewalls don't need support by the kernel, but
 - they are often combined with a packet filter, which only works if
 - you say Y here.
-
- You should also say Y here if you intend to use your Linux box as
 - the gateway to the Internet for a local network of machines without
 - globally valid IP addresses. This is called "masquerading": if one
 - of the computers on your local network wants to send something to
 - the outside, your box can "masquerade" as that computer, i.e. it
 - forwards the traffic to the intended outside destination, but
 - modifies the packets to make it look like they came from the
 - firewall box itself. It works both ways: if the outside host
 - replies, the Linux box will silently forward the traffic to the
 - correct local computer. This way, the computers on your local net
 - are completely invisible to the outside world, even though they can
 - reach the outside and can receive replies. It is even possible to
 - run globally visible servers from within a masqueraded local network
 - using a mechanism called portforwarding. Masquerading is also often
 - called NAT (Network Address Translation).
-
- Another use of Netfilter is in transparent proxying: if a machine on
 - the local network tries to connect to an outside host, your Linux
 - box can transparently forward the traffic to a local server,
 - typically a caching proxy server.
-
- Yet another use of Netfilter is building a bridging firewall. Using
 - a bridge with Network packet filtering enabled makes iptables "see"
 - the bridged traffic. For filtering on the lower network and Ethernet
 - protocols over the bridge, use ebttables (under bridge netfilter
 - configuration).
-
- Various modules exist for netfilter which replace the previous
 - masquerading (ipmasqadm), packet filtering (ipchains), transparent
 - proxying, and portforwarding mechanisms. Please see
 - <file:Documentation/Changes> under "iptables" for the location of
 - these packages.
-
- Make sure to say N to "Fast switching" below if you intend to say Y
- here, as Fast switching currently bypasses netfilter.

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

```
–  
– Chances are that you should say Y here if you compile a kernel which  
– will run as a router and N for regular hosts. If unsure, say N.  
–  
–if NETFILTER  
–  
–config NETFILTER_DEBUG  
– bool "Network packet filtering debugging"  
– depends on NETFILTER  
– help  
– You can say Y here if you want to get additional messages useful in  
– debugging the netfilter code.  
–  
–config BRIDGE_NETFILTER  
– bool "Bridged IP/ARP packets filtering"  
– depends on BRIDGE && NETFILTER && INET  
– default y  
– ---help---  
– Enabling this option will let arptables resp. iptables see bridged  
– ARP resp. IP traffic. If you want a bridging firewall, you probably  
– want this option enabled.  
– Enabling or disabling this option doesn't enable or disable  
– ebttables.  
–  
– If unsure, say N.  
–  
–source "net/ipv4/netfilter/Kconfig"  
–source "net/ipv6/netfilter/Kconfig"  
–source "net/decnnet/netfilter/Kconfig"  
–source "net/bridge/netfilter/Kconfig"  
–  
–endif  
–  
–config XFRM  
– bool  
– depends on NET  
–  
–source "net/xfrm/Kconfig"  
–  
source "net/sctp/Kconfig"  
  
config ATM  
@@ -294,50 +219,6 @@ config ATM_BR2684_IPFILTER  
    large number of IP-only vcc's. Do not enable this unless you are sure  
    you know what you are doing.  
  
–config BRIDGE  
– tristate "802.1d Ethernet Bridging"  
– ---help---  
– If you say Y here, then your Linux box will be able to act as an  
– Ethernet bridge, which means that the different Ethernet segments it
```

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

- is connected to will appear as one Ethernet to the participants.
- Several such bridges can work together to create even larger
- networks of Ethernets using the IEEE 802.1 spanning tree algorithm.
- As this is a standard, Linux bridges will cooperate properly with
- other third party bridge products.
-
- In order to use the Ethernet bridge, you'll need the bridge
- configuration tools; see <file:Documentation/networking/bridge.txt>
- for location. Please read the Bridge mini–HOWTO for more
- information.
-
- If you enable iptables support along with the bridge support then you
- turn your bridge into a bridging IP firewall.
- iptables will then see the IP packets being bridged, so you need to
- take this into account when setting up your firewall rules.
- Enabling arptables support when bridging will let arptables see
- bridged ARP traffic in the arptables FORWARD chain.
-
- To compile this code as a module, choose M here: the module
- will be called bridge.
-
- If unsure, say N.
-
- config VLAN_8021Q
- tristate "802.1Q VLAN Support"
- ----help----
- Select this and you will be able to create 802.1Q VLAN interfaces
- on your ethernet interfaces. 802.1Q VLAN supports almost
- everything a regular ethernet interface does, including
- firewalling, bridging, and of course IP traffic. You will need
- the 'vconfig' tool from the VLAN project in order to effectively
- use VLANs. See the VLAN web page for more information:
- <<http://www.candelatech.com/~greear/vlan.html>>
-
- To compile this code as a module, choose M here: the module
- will be called 8021q.
-
- If unsure, say N.
-
- config DECNET
- tristate "DECnet Support"
- help----
- @@ –479,32 +360,6 @@ config LAPB
- To compile this driver as a module, choose M here: the
- module will be called lapb. If unsure, say N.
-
- config NET_DIVERT
- bool "Frame Diverter (EXPERIMENTAL)"
- depends on EXPERIMENTAL
- ----help----
- The Frame Diverter allows you to divert packets from the

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

- network, that are not aimed at the interface receiving it (in
- promisc. mode). Typically, a Linux box setup as an Ethernet bridge
- with the Frames Diverter on, can do some *really* transparent www
- caching using a Squid proxy for example.
-
- This is very useful when you don't want to change your router's
- config (or if you simply don't have access to it).
-
- The other possible usages of diverting Ethernet Frames are
- numerous:
- – reroute smtp traffic to another interface
- – traffic–shape certain network streams
- – transparently proxy smtp connections
- – etc...
-
- For more informations, please refer to:
- <<http://diverter.sourceforge.net/>>
- <<http://perso.wanadoo.fr/magpie/EtherDivert.html>>
-
- If unsure, say N.
-

config ECONET

tristate "Acorn Econet/AUN protocols (EXPERIMENTAL)"

depends on EXPERIMENTAL && INET

@@ –538,6 +393,180 @@ config ECONET_NATIVE

Say Y here if you have a native Econet network card installed in your computer.

+source "net/ax25/Kconfig"

+

+source "net/irda/Kconfig"

+

+source "net/bluetooth/Kconfig"

+

+endmenu

+# end options and protocols

+

+menuconfig NETFILTER

+ bool "Network packet filtering (replaces ipchains)"

+ ---help---

+ Netfilter is a framework for filtering and mangling network packets

+ that pass through your Linux box.

+

+ The most common use of packet filtering is to run your Linux box as

+ a firewall protecting a local network from the Internet. The type of

+ firewall provided by this kernel support is called a "packet

+ filter", which means that it can reject individual network packets

+ based on type, source, destination etc. The other kind of firewall,

+ a "proxy–based" one, is more secure but more intrusive and more

+ bothersome to set up; it inspects the network traffic much more

+ closely, modifies it and has knowledge about the higher level

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

+ protocols, which a packet filter lacks. Moreover, proxy–based
+ firewalls often require changes to the programs running on the local
+ clients. Proxy–based firewalls don't need support by the kernel, but
+ they are often combined with a packet filter, which only works if
+ you say Y here.

+
+ You should also say Y here if you intend to use your Linux box as
+ the gateway to the Internet for a local network of machines without
+ globally valid IP addresses. This is called "masquerading": if one
+ of the computers on your local network wants to send something to
+ the outside, your box can "masquerade" as that computer, i.e. it
+ forwards the traffic to the intended outside destination, but
+ modifies the packets to make it look like they came from the
+ firewall box itself. It works both ways: if the outside host
+ replies, the Linux box will silently forward the traffic to the
+ correct local computer. This way, the computers on your local net
+ are completely invisible to the outside world, even though they can
+ reach the outside and can receive replies. It is even possible to
+ run globally visible servers from within a masqueraded local network
+ using a mechanism called portforwarding. Masquerading is also often
+ called NAT (Network Address Translation).

+
+ Another use of Netfilter is in transparent proxying: if a machine on
+ the local network tries to connect to an outside host, your Linux
+ box can transparently forward the traffic to a local server,
+ typically a caching proxy server.

+
+ Yet another use of Netfilter is building a bridging firewall. Using
+ a bridge with Network packet filtering enabled makes iptables "see"
+ the bridged traffic. For filtering on the lower network and Ethernet
+ protocols over the bridge, use ebttables (under bridge netfilter
+ configuration).

+
+ Various modules exist for netfilter which replace the previous
+ masquerading (ipmasqadm), packet filtering (ipchains), transparent
+ proxying, and portforwarding mechanisms. Please see
+ <file:Documentation/Changes> under "iptables" for the location of
+ these packages.

+
+ Make sure to say N to "Fast switching" below if you intend to say Y
+ here, as Fast switching currently bypasses netfilter.

+
+ Chances are that you should say Y here if you compile a kernel which
+ will run as a router and N for regular hosts. If unsure, say N.

+
+if NETFILTER
+
+config NETFILTER_DEBUG
+ bool "Network packet filtering debugging"
+ depends on NETFILTER
+ help

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

- + You can say Y here if you want to get additional messages useful in debugging the netfilter code.
- +
 - +config BRIDGE_NETFILTER
 - + bool "Bridged IP/ARP packets filtering"
 - + depends on BRIDGE && NETFILTER && INET
 - + default y
 - + ---help---
 - + Enabling this option will let arptables resp. iptables see bridged ARP resp. IP traffic. If you want a bridging firewall, you probably want this option enabled.
 - + Enabling or disabling this option doesn't enable or disable ebttables.
- +
 - + If unsure, say N.
 - +
 - +source "net/ipv4/netfilter/Kconfig"
 - +source "net/ipv6/netfilter/Kconfig"
 - +source "net/decnnet/netfilter/Kconfig"
 - +source "net/bridge/netfilter/Kconfig"
 - +endif
 - +# endif NETFILTER
- +
 - +config XFRM
 - + bool
 - + depends on NET
 - +
 - +source "net/xfrm/Kconfig"
- +
 - +config BRIDGE
 - + tristate "802.1d Ethernet Bridging"
 - + ---help---
 - + If you say Y here, then your Linux box will be able to act as an Ethernet bridge, which means that the different Ethernet segments it is connected to will appear as one Ethernet to the participants.
 - + Several such bridges can work together to create even larger networks of Ethernets using the IEEE 802.1 spanning tree algorithm.
 - + As this is a standard, Linux bridges will cooperate properly with other third party bridge products.
 - +
 - + In order to use the Ethernet bridge, you'll need the bridge configuration tools; see <file:Documentation/networking/bridge.txt> for location. Please read the Bridge mini–HOWTO for more information.
 - +
 - + If you enable iptables support along with the bridge support then you turn your bridge into a bridging IP firewall.
 - + iptables will then see the IP packets being bridged, so you need to take this into account when setting up your firewall rules.
 - + Enabling arptables support when bridging will let arptables see

Linux–Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

```
+ bridged ARP traffic in the arptables FORWARD chain.
+
+ To compile this code as a module, choose M here: the module
+ will be called bridge.
+
+ If unsure, say N.
+
+config VLAN_8021Q
+ tristate "802.1Q VLAN Support"
+ ----help----
+ Select this and you will be able to create 802.1Q VLAN interfaces
+ on your ethernet interfaces. 802.1Q VLAN supports almost
+ everything a regular ethernet interface does, including
+ firewalling, bridging, and of course IP traffic. You will need
+ the 'vconfig' tool from the VLAN project in order to effectively
+ use VLANs. See the VLAN web page for more information:
+ <http://www.candelatech.com/~greear/vlan.html>
+
+ To compile this code as a module, choose M here: the module
+ will be called 8021q.
+
+ If unsure, say N.
+
+config NET_DIVERT
+ bool "Frame Diverter (EXPERIMENTAL)"
+ depends on EXPERIMENTAL
+ ----help----
+ The Frame Diverter allows you to divert packets from the
+ network, that are not aimed at the interface receiving it (in
+ promisc. mode). Typically, a Linux box setup as an Ethernet bridge
+ with the Frames Diverter on, can do some *really* transparent www
+ caching using a Squid proxy for example.
+
+ This is very useful when you don't want to change your router's
+ config (or if you simply don't have access to it).
+
+ The other possible usages of diverting Ethernet Frames are
+ numerous:
+ - reroute smtp traffic to another interface
+ - traffic–shape certain network streams
+ - transparently proxy smtp connections
+ - etc...
+
+ For more informations, please refer to:
+ <http://diverter.sourceforge.net/>
+ <http://perso.wanadoo.fr/magpie/EtherDivert.html>
+
+ If unsure, say N.
+
+config WAN_ROUTER
+   tristate "WAN router"
```

Linux-Kernel: [RFC/PATCH] network configs: disconnect network options from drivers

```
depends on EXPERIMENTAL
@@ -605,6 +634,7 @@ config NET_SCHED
source "net/sched/Kconfig"

endmenu
+# end SCHED

menu "Network testing"

@@ -624,32 +654,8 @@ config NET_PKTGEN
    module will be called pktgen.

endmenu
+# end PKTGEN
```

```
endmenu
-
- config NETPOLL
- def_bool NETCONSOLE
-
- config NETPOLL_RX
- bool "Netpoll support for trapping incoming packets"
- default n
- depends on NETPOLL
-
- config NETPOLL_TRAP
- bool "Netpoll traffic trapping"
- default n
- depends on NETPOLL
-
- config NET_POLL_CONTROLLER
- def_bool NETPOLL
-
- source "net/ax25/Kconfig"
-
- source "net/irda/Kconfig"
-
- source "net/bluetooth/Kconfig"
-
- source "drivers/net/Kconfig"
-
- endmenu
+# end top support: options and protocols
```

~Randy

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>