# Linux TIPC 1.7 User's Guide

15 April 2010 [software version: TIPC 1.7.7, tipc-config 1.1.8]

## Table of Contents

### Introduction

This document is provided to assist software developers in setting up and operating a network using Linux TIPC 1.7.

For more information about the TIPC protocol, including information about writing applications that use TIPC, please consult the open source TIPC project website at http://tipc.sf.net/index.html . This site contains TIPC project software, documentation, news, and support instructions.

The TIPC development team welcomes input from the TIPC user community! Feel free to provide feedback on TIPC using the normal TIPC support procedures outlined at http://tipc.sf.net/support.html .

# 1. What's New

TIPC 1.7.7 contains only minor changes from the previous TIPC 1.7.6 release:

- adds support for Linux 2.6.29-2.6.34 kernels
- adds support for four new socket options
- fixes a number of bugs present in TIPC 1.7.6

TIPC 1.7.6 contains a number of significant advances from earlier TIPC 1.5/1.6 releases:

- adds initial support for hierarchical networks (i.e. multi-cluster and multi-zone)
- adds memory footprint reduction options
- incorporates a variety of socket API improvements
- incorporates a variety of performance improvements
- fixes a variety of bugs present in TIPC 1.5/1.6

Information on the new features of TIPC can be found in Section 8, “New Features” . For information on all other features of TIPC, please consult the TIPC 1.5/1.6 documentation available at http://tipc.sf.net/documentation.html .

> *Note:* A complete record of significant changes in each version of TIPC can be found in the TIPC change history ( http://tipc.sourceforge.net/history.html ).

# 2. Release Status

TIPC 1.7 is currently in maintenance mode. All features targeted for this release have now been integrated, and only minor enhancements and bug fixes are being incorporated.

Please report all issues and suggestions for improvements to the TIPC development team, using the normal TIPC support procedures outlined at http://tipc.sf.net/support.html .

# 3. System Requirements

TIPC 1.7.7 can be used with Linux kernel 2.6.16 through 2.6.34, so it will run on any system currently running TIPC 1.6.

> *Note:* Support for Linux kernels 2.6.9 through 2.6.14 is currently unavailable, so systems running TIPC 1.5 cannot run TIPC 1.7 without a kernel upgrade. (An unsupported patch that allows TIPC 1.6 to run on these kernels has been published previously and might work for TIPC 1.7, but this has not been verified.)

TIPC 1.7 requires tipc-config version 1.1.x for run-time configuration and operation.

The latest versions of TIPC 1.7 and its associated software are available at http://tipc.sf.net/download.html .

# 4. Installation

These instructions assume that your system is running Linux kernel 2.6.16 (or later), and that you are already familiar with the steps involved in rebuilding a "vanilla" Linux kernel from source provided at www.kernel.org.

> *Caution:* Some Linux distributions use a different procedure to build a kernel than the one described here, in which case you will need to adapt your actions accordingly.

If you aren't comfortable with the idea of updating your existing kernel, check to see if the kernel supports loadable modules; if so, you can build TIPC as a loadable module and install it dynamically.

1. Copy tipc-1.7.x.tar to the top level directory of the kernel source tree.
2. Install the TIPC 1.7 source files into the kernel source tree.

   ```
   eg. tar -xvf tipc-1.7.x.tar
   ```

   > *Note:* This replaces any existing TIPC 1.6 or 1.7 files with the new TIPC 1.7 files. Some obsolete TIPC files may remain, but they will be ignored when the TIPC module is (re)built.

3. Configure the kernel to include TIPC, either statically or as a loadable module.

```
eg. make menuconfig
```

The TIPC configuration menu is located at: Networking # Networking options # The TIPC Protocol (EXPERIMENTAL)

> *Note:* TIPC's default configuration settings should be sufficient for the needs of most users.

4. Rebuild and install the kernel in the normal manner.

```
eg. make
```

If you are building TIPC as a loadable module, build net/tipc/tipc.ko and install it in the standard manner.

> *Note:* The README file in the top level directory of a kernel source tree obtained from www.kernel.org provides a more complete description of how to build the Linux kernel and loadable modules.

5. Boot up your system, then build the tipc-config tool and use it to configure and manage TIPC.

```
eg. cd <tipc-config source directory>
    make
    ./tipc-config <commands>
```

> *Note:* Be sure to use a version 1.1.x release of tipc-config, as the earlier 1.0.x series may not operate correctly with TIPC 1.7.

# 5. Network Configuration

Most TIPC users currently configure single-cluster networks in which each networks node is connected to a common LAN using a single network interface per node. A smaller proportion of users utilize a second network interface per node to create an alternate, independent LAN, in order to provide redundancy in the event of a LAN failure.

To allow a new node running TIPC to join a single cluster TIPC network you simply use the **tipc-config** tool to tell the node the network ID and network address it should use, and to tell it which network interface(s) to use. For example, to assign TIPC the address <1.1.8> on network 1234 and have it use Ethernet interface eth0 you would enter:

```
tipc-config -netid=1234 -a=1.1.8 -be=eth:eth0
```

You can confirm that the interface(s) have been successfully enabled by examining the Linux kernel log. (For example, use **dmesg** or look at /var/log/messages.) If there are other TIPC nodes with the same network ID present within your LAN you will also see printouts showing links being established.

> *Caution:* If you want to use redundant LANs it is very important that they are not interconnected. The auto-configuration mechanism will become utterly confused if a network interface used by TIPC receives neighbour detection broadcasts from a single

> neighboring node using two or more different Ethernet addresses in a round-robin manner. (This is essentially the same confusion that results if multiple nodes are assigned the same <Z.C.N> address.)

Many users will have no need to do any additional configuration of TIPC, since the system defaults will be sufficient for their purposes. Some users with special needs may need to take additional steps.

- Users of multi-cluster or multi-zone networks should read Section 8.1, “Hierarchical networks” to learn about special steps that may be required to configure their network.
- Users who wish to change the default limits used by TIPC (eg. maximum # of publications permitted) can do so using the Linux kernel configuration tool and then rebuilding TIPC. (Most of these limits can also be changed without rebuilding TIPC using **tipc-config** commands, although there are limitations that restrict when such commands may be issued.)

A more detailed description of all network configuration commands provided by **tipc-config** can be found in Section 7, “Command Reference”.

# 6. Network Monitoring

The **tipc-config** tool allows users to monitor the behavior of a TIPC network. Commonly used commands include the following:

```
    -l      - display status of all links created by TIPC
    -n      - display status of all neighboring nodes discovered by TIPC
    -nt     - display all name table information used by TIPC
    -p      - display all ports created by TIPC
```

A more detailed description of all network monitoring commands provided by **tipc-config** can be found in Section 7, “Command Reference”.

# 7. Command Reference

The **tipc-config** tool provides numerous commands for configuring and monitoring a TIPC network. Certain commands are only available to users having network administrator privileges (eg. root) to prevent unauthorized users from reconfiguring the network.

Users can pass multiple commands to **tipc-config** in a single invocation. Commands are normally processed serially, from left to right; exceptions are the "-v", "-i", and "-dest" commands, which have an immediate effect on all commands in a command set. If **tipc-config** detects a failure while executing a set of commands, it exits without attempting any unprocessed commands.

Command names and arguments are case-sensitive. Commands can be abbreviated by truncating the command name, as long as the abbreviation is unambiguous. For example, "-addr" may be abbreviated to "-a" or "-ad", but "-max_zones" cannot be abbreviated to "-max".

> *Note:* A full command name is never ambiguous, even if it is a substring of another command. For example, "-m" is non-ambiguous, even though there are other commands that begin with the same sequence of characters (eg. "-mng").

This command reference utilizes the following syntax conventions:

```
<addr>       - a network node (eg. 1.1.12)
<domain>     - a network region (eg. 2.3.12, 2.3.0, 2.0.0, or 0.0.0)
<linkname>   - a link name (eg. 1.1.10:eth3-1.1.17:eth2)
<linkpat>    - a link name pattern (eg. <linkname> or ?1.1.17 or ?eth3)
<bearer>     - a bearer name (eg. eth:eth0)
<bearerpat>  - a bearer name pattern (eg. <bearer> or ?eth)
<media>      - a media type (eg. eth)
```

A pattern argument can be either a full name or a partial name (denoted by an initial '?' character). A partial name matches all names containing the string that follows the '?'.

Any command argument that is not described by the conventions above denotes an unsigned integer value.

The following commands are supported:

## -addr[=<addr>]

Set the network address of the node. If <addr> is omitted, the node's current network address is displayed.

> *Note:* Once the node's network address has been set, it is no longer possible to alter certain TIPC configuration settings, such as the network ID and certain "-max_XXX" limits.

## -b[=<bearerpat>]

List all bearers having a name that matches the specified pattern. If no pattern is specified, all bearers are listed.

## -bd=<bearerpat>

Disable the bearers having a name that matches the specified pattern.

## -be=<bearer>[/<domain>[/<priority>]]

Enable a TIPC bearer on the interface <bearer>. The bearer name has the form <media>:<ifname>, indicating the media type used by the interface and the interface's name (eg. eth:eth0). More than one interface may be enabled at a time by specifying multiple bearers in a comma separated list (e.g. -be=eth:eth0,eth:eth1).

Once enabled, TIPC starts broadcasting over the bearer to detect other nodes in its network. The optional <domain> value specifes a "neighbour detection domain" which limits the nodes that TIPC can set up links to. For example, <0.0.0> (the default) tells TIPC to set up links to all nodes it finds, while <1.1.0> tells TIPC to communicate only with nodes that are part of cluster <1.1>.

The optional <priority> value specifies the priority for any TIPC links created by the bearer. Priorities can range from 0 (lowest) to 31 (highest); a priority of 32 (the default) tells TIPC to use the standard priority associated with the bearer's media type.

## -dest=<addr>

Perform management commands on node <addr>. Other commands issued as part of the same command set are directed to the specified node for processing. (Note: Some commands that alter the configuration of TIPC cannot be performed remotely.)

## -help

Display a summary of the options supported by **tipc-config**.

## -i

Enable "interactive" mode. Other commands issued as part of the same command set will prompt for confirmation before attempting to change the configuration of TIPC.

## -l[=<domain>|<linkpat>]

List all links to neighboring nodes in the specified network domain, or whose name matches the specified pattern.

If no argument is supplied, <domain> defaults to <0.0.0>, which causes the links to all neighboring nodes to be listed.

## -log[=<size>]

Set the size of TIPC's system log to the specified number of bytes. To disable logging of system messages specify a <size> of zero.

If <size> is omitted the contents of the system log are displayed, and the log is reset to empty.

## -lp=<linkpat>|<bearer>|<media>/<value>

Set the priority for links having a name that matches <linkpat> to <value>. Priorities can range from 0 (lowest) to 31 (highest).

This command can also be used to change the default priority assigned to new links created by the specified bearer (eg. -lp=eth:eth1/<value>), or to links created by new bearers of the specified media type (eg. -lp=eth/<value>). Note that such use will have no effect on the priority of existing links or the default priority of existing bearers.

If TIPC has more than one link to the same neighboring node, it will send traffic over the links with the highest priority, and only utilize lower priority links if all higher priority links have failed. If there are two links having highest priority, TIPC sends traffic over both links (although not necessarily in equal amounts).

## -ls[=<linkpat>]

Display status and statistics information for links having a name that matches the specified pattern. If no pattern is specified, information for all links is displayed.

## -lsr=<linkpat>

Reset the statistics counters for links having a name that matches the specified pattern.

## -lt=<linkpat>|<bearer>|<media>/<value>

Set the tolerance for links having a name that matches <linkpat> to <value> milliseconds.

This command can also be used to change the default tolerance assigned to new links created by the specified bearer (eg. -lt=eth:eth1/<value>), or to links created by new bearers of the specified media type (eg. -lt=eth/<value>). Note that such use will have no effect on the tolerance of existing links or the default tolerance of existing bearers.

Link tolerance specifies the maximum time that TIPC will allow a communication problem to exist before taking the link down.

# -lw=<linkpat>|<bearer>|<media>/<value>

Set the window for links having a name that matches <linkpat> to <value> messages.

This command can also be used to change the default window assigned to new links created by the specified bearer (eg. -lw=eth:eth1/<value>), or to links created by new bearers of the specified media type (eg. -lw=eth/<value>). Note that such use will have no effect on the window of existing links or the default window of existing bearers.

The link window controls how many unacknowledged messages a link endpoint can have in its transmit queue before TIPC's congestion control mechanisms become active.

# -m

List all media types supported by TIPC.

# -max_clusters[=<value>]

Set the maximum number of clusters supported by the node's zone to <value>. If <value> is omitted the current setting is displayed.

> *Note:* Once the node's network address has been set it is no longer possible to alter this setting.

# -max_nodes[=<value>]

Set the maximum number of nodes supported by the node's cluster to <value>. If <value> is omitted the current setting is displayed.

> *Note:* Once the node's network address has been set it is no longer possible to alter this setting.

# -max_ports[=<value>]

Set the maximum number of ports supported by the node to <value>. If <value> is omitted the current setting is displayed.

> *Note:* TIPC does not yet allow this setting to be changed once TIPC has been activated. Users wishing to change the default setting must reconfigure the Linux kernel and rebuild TIPC.

# -max_publ[=<value>]

Set the maximum number of name publications (eg. socket bind() operations) supported by the node to <value>. If <value> is omitted the current setting is displayed.

# -max_remotes[=<value>]

Set the maximum number of neighboring nodes in other clusters to <value>. If <value> is omitted the current setting is displayed.

> *Note:* Once the node's network address has been set it is no longer possible to alter this setting.

# -max_subscr[=<value>]

Set the maximum number of name subscriptions supported by the node to <value>. If <value> is omitted the current setting is displayed.

# -max_zones[=<value>]

Set the maximum number of zones supported by the node's network to <value>. If <value> is omitted the current setting is displayed.

> *Note:* Once the node's network address has been set it is no longer possible to alter this setting.

# -mng[=enable|disable]

Permit or disallow processing of configuration commands issued by other nodes.

# -n[=<domain>]

List all neighboring nodes within the specified network domain. If <domain> is omitted it defaults to <0.0.0>, which causes all neighboring nodes to be listed.

# -netid[=<value>]

Set the network ID of the node to <value>. If <value> is omitted the current network ID is displayed.

This command makes it possible to configure multiple independent networks on a LAN whose nodes will not interact with each other. If you are the only TIPC user on the LAN, you can use the default network ID.

> *Note:* Once the node's network address has been set it is no longer possible to alter the network ID.

# -nt[=[<depth>,]<type>[,<low>[,<up>]]]

Display information from TIPC's name table.

Use <depth> to control how much detail is displayed for each entry listed:

```
types = displays type info only
names = displays type and instance info (i.e. name info)
ports = displays type, instance, and port info
all   = displays type, instance, port, and publication info
```

If <depth> is omitted, it defaults to "all".

Use <type>,<low>,<up> to control how many name table entries are displayed:

```
  <type>            displays all entries for the specified type
  <type>,<low>      displays all entries overlapping the specified name
  <type>,<low>,<up> displays all entries overlapping the specified name sequence
```

If <type> is omitted, all name table entries are displayed.

### -p

List all ports created by TIPC on the node.

### -r[=<domain>]

List all known routes to the specified network domain. If <domain> is omitted it defaults to <0.0.0>, which causes all known routes to be listed.

Each route listed consists of three network addresses:

- Region: a cluster (<Z.C.0>) or zone (<Z.0.0>) that can be reached from this node
- Local router: a node in this node's cluster/zone that has a direct link to that region
- Remote router: a node in the destination cluster/zone that completes the direct link to that region

### -s

Display TIPC status information. Currently, the only information provided is the version of TIPC being used (eg. TIPC version 1.7.x).

### -v

Enable "verbose" mode. Other commands issued as part of the same command set may produce more detailed output describing what they are doing.

### -V

Display the version of **tipc-config** being used.

# 8. New Features

This section describes new features or enhancements in TIPC 1.7 that impact the use of TIPC.

## 8.1. Hierarchical networks

TIPC 1.7 introduces support for networks containing multiple clusters in a zone, and multiple zones in a network. This allows a node to interact with nodes to which it does not have a direct link.

To create a multi-cluster network, do the following:

- assign each node with a unique network address (i.e. <Z.C.N> value)

- ensure each node in a cluster has a link to every other node in its cluster
- ensure each cluster in a zone has at least one link to every other cluster in that zone

For example, a user can create the following logical network comprising two clusters of three nodes each. (Intra-cluster links are denoted using '-', while inter-cluster links are denoted by '='.)

```
|---<1.1.1>=======<1.2.1>---|
|                           |
|---<1.1.2>       <1.2.2>---|
|                           |
|---<1.1.3>       <1.2.3>---|
```

In such a network, applications running on any node in cluster <1.1> can detect and exchange messages with applications running on any node in cluster <1.2>, providing those applications were created with "zone" scope. TIPC will then route messages between any pair of node by the shortest available path.

To create a multi-zone network, users must do the following:

- create two or more zones, each containing one or more clusters, as described above
- ensure each zone in the network has at least one link to every other zone in the network

For example, a user can extend the previous example by adding in a second two-cluster zone which has smaller clusters. (The new inter-zone link is denoted by '#'.)

```
|---<2.1.12>     <2.2.17>---|
|                           |
|---<2.1.25>=====<2.2.12>---|
       #
       #
       #
|---<1.1.1>=======<1.2.1>---|
|                           |
|---<1.1.2>       <1.2.2>---|
|                           |
|---<1.1.3>       <1.2.3>---|
```

> *Note:* TIPC currently does not support the exchange of name information between different zones. Consequently, an application can only communicate with another application residing in a different zones if it knows which zone that application resides in. Support for inter-zone name distribution may be added in a future release.

### 8.1.1. Assigning network addresses

Restrictions on network addresses have been loosened in TIPC 1.7, making it possible to assign an arbitrary <Z.C.N> value to a TIPC node.

```
eg. tipc-config -netid=1234 -a=1.3.253
```

### 8.1.2. Configuring links

When setting up a node to be part of a hierarchical network, the network administrator must assign a specific network "domain" to each interface used by the node to indicate which of its neighbouring nodes are allowed to establish links through the interface. The domain specifies a range of <Z.C.N> values, and operates something along the lines of the subnet mask used in IP networks.

For example, to configure interface eth0 on node <1.2.5> to allow links within the node's own cluster (i.e. to any node with address <1.2.x>):

```
tipc-config -be=eth:eth0/1.2.0
```

To configure interface eth0 on node <1.2.5> to allow links within the node's own zone (i.e. to any node with address <1.x.y>):

```
tipc-config -be=eth:eth0/1.0.0
```

Lastly, to configure interface eth0 on node <1.2.5> to allow links to any node in the network:

```
tipc-config -be=eth:eth0/0.0.0
```

If the domain value is omitted tipc-config configures the interface to establish intra-cluster links only.

As might be expected, TIPC will only create a link between two nodes if *both* nodes connected to the interface are configured to allow communication with the other node.

### 8.1.3. Selecting a network topology

It is important to understand that the topology of a TIPC network is a logical construct that may differ from the topology of the underlying physical network. While TIPC requires that a physical connection exist between two nodes before it can establish a logical link between them, the network administrator has total control over which physical connections TIPC will use to establish these links. This means that it may be possible to create a given logical network topology using two or more different physical network topologies.

For example, the first logical network shown at the start of this section can be implemented using the following physical topology:

```
 |---<1.1.1>===|===<1.2.1>---|
 |             |             |
 |---<1.1.2>XXX|XXX<1.2.2>---|
 |             |             |
 |---<1.1.3>XXX|XXX<1.2.3>---|
eth0          eth1          eth2
```

where 'XXX' indicates that links are not established over an interface even though the node is physically connected to it. The network administrator achieves this effect by configuring all nodes to establish intra-cluster links over one interface (either "eth0" or "eth2"), and configuring <1.1.1> and <1.2.1> to also use "eth1" for inter-cluster links.

However, the network could also be implemented using the following single Ethernet segment physical topology.

```
<1.1.1>===|===<1.2.1>
      \___|___/
          |
<1.1.2>---|---<1.2.2>
          |
<1.1.3>---|---<1.2.3>
         eth0
```

In this case, all nodes are configured to use "eth0", but only nodes <1.1.1> and <1.2.1> are permitted to establish inter-cluster links.

TIPC has been designed with the intent of supporting a wide range of network topolgies, both physical and logical. The network architect must determine the best topology on a case-by-case basis to ensure the needs of the applications running in the network are met. However, the following guidelines should be noted.

- The logical TIPC network must ensure that each node in a cluster has a link to every other node in its cluster, that each cluster in a zone has at least one link to every other cluster in that zone, and that each zone has at least one link to every other zone.

- The creation of redundant paths between clusters and zones should be used with caution.

  If redundant paths between clusters are configured, the clusters must provide redundancy for their intra-cluster links as well; similarly, if redundant paths between zones are configured, the zones must provide redundant paths between their own clusters. Failure to adhere to this requirement makes it easier for a node or link failure condition to result in an unsupported network topology (i.e. one in which TIPC's full meshing requirements are not met), leading to unexpected and/or undesired network behavior.

  Even when sufficient link/path redundancy is provided, it is possible for a change in the logical network topology (such as the addition or failure of a node or link) to result in inter-cluster messages (or inter-zone messages) arriving out-of-order at their destination, which may be particularly undesirable for connection-oriented traffic.

  Both of these concerns can be avoided by ensuring that only a single path is configured between any two clusters and/or zones.

## 8.2. Reducing footprint

New TIPC kernel configuration options allow users to omit certain TIPC capabilities from the TIPC kernel module if they are not required. These options may be useful when running TIPC on a resource-constrained system with limited memory.

Capabilities that may be omitted include:

- Uni-cluster TIPC support, which allows TIPC to inter-operate with nodes in its cluster running TIPC 1.5/1.6. (For more information on this subject, see Section 9, “Interoperability” .) This capability can be excluded if all nodes in the TIPC network are running TIPC 1.7.

- Support for redundant links, which allows TIPC to redirect messages to an alternate link (if one is available) in the event of link failure. This capability can be excluded if there is one interface by which TIPC can reach each of its neighboring nodes.

- TIPC's configuration service, which allows TIPC to be configured and monitored using the tipc-config tool. Currently this service is required if TIPC is to be used as part of a multi-node network, so users will need to include it unless they plan to run TIPC as an isolated node.

- TIPC's socket API, which allows TIPC applications to use sockets of the AF_TIPC address family. This capability can be excluded if TIPC is only required to support applications using TIPC's native API. Currently the tipc-config tool requires the use of TIPC sockets, so users will usually need to include this capability.

- TIPC system message support, which causes TIPC to record info about errors, warnings, and significant events (like link establishment and link failure) to the system console and TIPC's log buffer.

- TIPC debug message support, which allows TIPC to output info that helps in diagnosing problems. This capability requires the user to understand and modify TIPC's source code, and is intended for use by TIPC developers.

By default, all optional capabilities are included except for debug message support. These defaults can be changed using the Linux kernel configuration tool and then rebuilding TIPC.

## 8.3. Socket API changes

TIPC's socket API has undergone significant internal revision in TIPC 1.7. Although most changes are transparent to TIPC programmers, designers should take note of the following differences:

- TIPC now validates the "how" argument to **shutdown()** to help prevent accidental misuse of this routine.

  > *Important:* This change is NOT backwards compatible. Applications designed for TIPC 1.5/1.6 (or earlier versions) that used **shutdown(sock, 0)** must now use **shutdown(sock, SHUT_RDWR)** to ensure proper operation.

- Support for multi-threaded manipulation sockets has been enhanced. For example, it is now possible to have one thread of control send messages over a socket even when another thread is blocked trying to receive messages from that socket.

- Several performance improvements that enhance SOCK_STREAM data transfers may alter the amount of data returned by a receive operation, compared to the amount returned in previous versions of TIPC.

- Support for the SOL_SOCKET level socket option SO_RCVTIMEO has been added for all socket types. Also, support for SO_RCVLOWAT has been added for SOCK_STREAM sockets only.

- Support for the SOL_TIPC level socket options TIPC_SOCK_RECVQ_DEPTH and TIPC_NODE_RECVQ_DEPTH have been added for all socket types. These read-only options allow applications to determine the number of unprocessed messages that are sitting in the receive queues of TIPC sockets, which can be helpful in troubleshooting issues involving message congestion.

- A number of race conditions that could cause improper or unexpected behavior have been eliminated.

# 9. Interoperability

TIPC 1.7 is designed to interoperate with nodes running TIPC 1.5/1.6, as long as all nodes belong to the same cluster. This interoperability allows an existing TIPC 1.5/1.6 cluster to be expanded to include new nodes running TIPC 1.7, or to prepare the cluster for evolution into a multi-cluster TIPC network.

> *Important:* The TIPC development team makes no promise of interoperability when versions of TIPC 1.7 prior to TIPC 1.7.5 are used. (For example, TIPC 1.7.3 introduced an improved name and route distribution algorithm that is incompatible with earlier TIPC 1.7.x releases.) Users of TIPC 1.7 should ensure that all nodes in the network are running TIPC 1.7.5 or later.

To create a multi-cluster TIPC network, all nodes in the network must first be upgraded to TIPC 1.7. In the event that a TIPC 1.5/1.6 node is accidentally incorporated into a multi-cluster network, it will operate as if it is part of a single-cluster network. (That is, it will interact with the TIPC 1.7 and TIPC 1.5/1.6 nodes in its own cluster, and be ignored by all other nodes in the network.) However, all TIPC 1.7 nodes in the cluster will still be able to interact with the TIPC 1.7 nodes in other clusters in the normal multi-cluster manner.

It is recommended that the tipc-config tool only be used to remotely manage nodes running a compatible version of TIPC. (That is, tipc-config version 1.1.x should only be used to manage nodes running TIPC 1.7; likewise, tipc-config 1.0.x should only be used to manage TIPC 1.5/1.6 nodes.)

The TIPC development team makes no claims regarding the interoperability of a TIPC 1.7 node with a node running TIPC 1.3/1.4 software.

# 10. Known Issues

This release has been sanitized using the TIPC test suite, and runs the TIPC demos. All bugs present in this release are recorded in the TIPC bug tracker at http://sourceforge.net/tracker/index.php .

Please report all issues and suggestions for improvements to the TIPC development team using the normal TIPC support procedures outlined at http://tipc.sf.net/support.html .

[END OF DOCUMENT]