ETSI GR NGP 009 V1.1.1 (2019-02)

Next Generation Protocols (NGP); An example of a non-IP network protocol architecture based on RNA design principles

Disclaimer

The present document has been produced and approved by the Next Generation Protocols (NGP) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGR/NGP-009

Keywords
API, architecture, internet, meta-protocol, network, next generation protocol, protocol

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Next Generation Protocols (NGP).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

Network protocol architecture provides a set of patterns and methodology that guides network (protocol) designers in carrying out their task. It captures the rules and patterns that are invariant with respect to the specific requirements of each individual network (cellular, datacentre, sensor, access, core, enterprise, LAN, etc.). Today the prevalent network protocol architecture (usually referred to as "Internet protocol suite"), loosely based on OSI provides too little patterns and commonality and has fundamental design flaws in its structure, naming and addressing, service API and security. These issues contribute to an explosion in the number of the network protocols required, both to cover requirements of multiple use cases and to work around the fundamental design flaws.

The Recursive InterNetwork Architecture (RINA) is a "back to basics" approach learning from the experience with TCP/IP and other technologies in the past. Research results to date have found that many long-standing network problems can inherently be solved by the structure resulting from the theory of networking. Hence, additional mechanisms are not required.

RINA provides the adequate tools to solve the problems of the Internet architecture (complexity, scalability, security, mobility, quality of service or management to name a few). RINA is based on a single type of layer, which is repeated as many times as required by the network designer. The layer is called a Distributed IPC Facility (DIF), which is a distributed application that provides Inter Process Communication (IPC) services over a given scope to the distributed applications above (which can be other DIFs or regular applications). These IPC services are defined by the DIF API, which allows instances of applications -including other DIFs- to request IPC flows with certain characteristics (such as loss, delay, in-order delivery) to other application instances. Hence a layer can is a resource allocator that provides and manages the IPC service oven a given scope (link, network, internetwork, VPN, etc.). It allocates resources (memory in buffers, bandwidth, scheduling capacity) to competing flows.

All DIFs offer the same services through their API and have the same components and structure. Each layer features two sets of protocol frameworks: one for data transfer (called EFCP, Error and Flow Control Protocol), and one for layer management (CDAP, the Common Distributed Application Protocol). However, not all the DIFs operate over the same scope and environment nor do they have to provide the same level of service. Hence, invariant parts (mechanisms) and variant parts (policies) are separated in different components of the data transfer and layer management protocol frameworks. This makes it possible to customize the behaviour of a DIF to optimally operate in a certain environment with a set of policies for that environment instead of the traditional "one size fits all" approach or having to re-implement mechanisms in independent protocols over and over again.

Last but not least, RINA can be deployed incrementally where it has the right incentives, and interoperate with current technologies such as IP, Ethernet, MPLS, WiFi, Cellular or others.

1 Scope

Today most network protocols loosely follow the layering structure of the OSI network architecture. Protocols are organized in a static number of layers, in which each layer provides a different function to the layer above. The limitations of such structure have led to an explosion in the number of protocols at each layer with little or no commonality, layer violations and the need for ad-hoc extensions in the number of layers where the architecture could not model real-world networks with enough fidelity (e.g. layers 2,5 or 3,5, virtual networks, etc.). SDOs independently develop protocols for different layers of the protocol architecture, many times replicating each other's work and leading to inefficiencies at the system level. This results in:

a) networks that are highly complex to operate and troubleshoot;

b) specification and implementation of new protocols which add little value to the existing base; and

c) an overall networked system that is far from an optimal integration level from a systems design perspective.

The present document discusses the properties of a non-IP network architecture based on RINA design principles. Network architecture captures all the rules and patterns that are independent of the requirements addressed by individual network protocols. It solves the problems that are generic to any network (e.g. structure, naming and addressing, security models or QoS) at the architecture level, avoiding the need for individual protocols to solve these problems by themselves. RINA has been designed to capture the invariants of all forms of networking, providing SDOs and network designers with a common framework and methodology to design and build protocols for any type of network. Thus a network protocol architecture like RINA encourages networks with fewer protocols and more commonality, more cooperation between SDOs and simpler and more predictable networks.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 62: "A System for Inter-Process Communication in a Resource Sharing Network", August 1970, D.C. Walden.

NOTE: Available at https://tools.ietf.org/rfc/rfc62.txt.

[i.2] INWG-96 (1975): "Proposal for an International End To End Protocol", V. Cerf, A. McKenzie, R. Scantleburie, H. Zimmerman.

NOTE: Available at http://dotat.at/tmp/INWG-96.pdf.

[i.3] IETF RFC 793: "Transmission Control Protocol", September 1981, University of Southern California.

NOTE: Available at https://tools.ietf.org/html/rfc793.

[i.4] ETSI GS NGP 007: "Next Generation Protocols (NGP); NGP Reference Model".

NOTE: Available at https://www.etsi.org/deliver/etsi_gs/NGP/001_099/007/01.01.01_60/gs_NGP007v010101p.pdf.

[i.5] ISO/IEC 7498-1:1994: "Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model".

[i.6] IETF draft-ietf-taps-arch-02: "An Architecture for Transport Services", January 2018, T. Pauly, B. Trammell, A. Brunstrom, G. Fairhurst, C. Perkins, P. Tiesel, C. Wood.

NOTE: Available at https://datatracker.ietf.org/doc/draft-ietf-taps-arch/?include_text=1.

[i.7] P. B. Hansen: "The Nucleus of a Multi Programming System", Communications of the ACM 13 (4): 238-241. April 1970.

NOTE: Available at https://dl.acm.org/citation.cfm?id=362278&dl=ACM&coll=GUIDE.

[i.8] J. Day: "Patterns in Network Architecture: A return to Fundamentals". Prentica Hall, 2008.

[i.9] R. Watson: "Timer-based mechanism in reliable transport protocol connection management", Computer Networks, 5:47-56, 1981.

[i.10] G. Gursun, I. Matta and K. Mattar: "On the Performance and Robustness of Managing Reliable Transport Connections", 8^{th} International Workshop on PFLDNeT, November 2010.

[i.11] G. Boddapati, J. Day, I. Matta, L. Chitkushev: "Assessing the security of a clean-slate Internet architecture", 20^{th} IEEE conference on Network Protocols, 2012.

[i.12] E. Grasa, O. Rysavy, O. Lichtner, H. Asgari, J. Day, L. Chitkushev: "From protecting protocols to protecting layers: designing, implementing and experimenting with security policies in RINA", IEEE ICC 2016.

[i.13] IETF RFC 4291: "IPv6 addressing architecture", February 2006, R. Hinden, S. Deering.

NOTE: Available at https://tools.ietf.org/html/rfc4291.

[i.14] IETF RFC 4192: "Procedures for renumbering an IPv6 network without a flag day", September 2005, F. Baker, E. Lear, and R. Droms.

NOTE: Available at https://tools.ietf.org/html/rfc4192.

[i.15] IETF RFC 5887: "Renumbering still needs work", May 2010, B. Carpenter, R. Atkinson, and H. Flinck.

NOTE: Available at https://tools.ietf.org/html/rfc5887.

[i.16] D. Leroy and O. Bonaventure: "Preparing network configurations for IPv6 renumbering," International Journal of Network Management, vol. 19, no. 5, pp. 415-426, September/October 2009.

[i.17] J. Small: "Threat analysis of recursive internetwork architecture distributed IPC facilities", BU Technical report, 2011.

NOTE: Available at http://pouzinsociety.org/research/publications.

[i.18] Eduard Grasa, Leonardo Bergesio, Miquel Tarzan, Diego Lopez, John Day and Lou Chitkushev: "Seamless Network Renumbering in RINA: Automate Address Changes Without Breaking Flows!", EUCNC 2017.

NOTE: Available at https://zenodo.org/record/1013204#.WeTDrROCxTY.

[i.19] IETF RFC 5944: "IP mobility support for IPv4, revised", November 2010, C. Perkins.

NOTE: Available at https://tools.ietf.org/html/rfc5944.

[i.20] IETF RFC 6275: "Mobility support in IPv6", July 2011, J. A. C. Perkins, D. Johnson.

NOTE: Available at https://tools.ietf.org/html/rfc6275.

[i.21] IETF RFC 5213: "Proxy mobile IPv6", August 2008, S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury and B. Patil.

NOTE: Available at https://tools.ietf.org/html/rfc5213.

[i.22] IETF RFC 6830: "The Locator/ID Separation Protocol (LISP)", January 2013, D. Meyer, D. Lewis, D. Farinacci, V. Fuller.

[i.23] J. Day and E. Grasa: "Mobility Made Simple". PSOC Tutorial paper, May 2016.

NOTE: Available at http://psoc.i2cat.net/sites/default/files/PSOC-tutorial-Mobility-made-simple.pdf?_ga=2.45065702.356832367.1537337692-718608749.1512423093.

[i.24] V. Ishakian, J. Akinwumi, F. Esposito and I. Matta: "On supporting mobility and multihoming in recursive internet architectures", Comput. Commun., vol. 35, no. 13, pp. 1561-1573, July 2012.

[i.25] ETSI GS NGP 001 (V1.3.1): "Next Generation Protocols (NGP); Scenario Definitions".

NOTE: Available at https://www.etsi.org/deliver/etsi_gs/NGP/001_099/001/01.03.01_60/gs_NGP001v010301p.pdf.

[i.26] IETF RFC 1287: "Towards the Future Internet Architecture", December1991, D. Clark, L. Chapin, V. Cerf, R. Braden, R. Hobby.

NOTE: Available at https://tools.ietf.org/html/rfc1287.

[i.27] J. Day: "How in the heck do you lose a layer!?", International Conference on the Network of the Future, 2011.

[i.28] Atlantic Council, Frederik S. Pardee Center for International Futures, Zurich: "Risk Nexus. Overcome by Cyber Risks? Economic benefits and costs of alternate cyber futures", 2016.

NOTE: Available at http://publications.atlanticcouncil.org/cyberrisks//.

[i.29] IETF RFC 2535: "Domain name system security extensions", March 1999, D. Eastlake.

NOTE: Available at https://tools.ietf.org/html/rfc2535.

[i.30] IETF RFC 2401: "Security architecture for the IP Protocol", December 2005, S. Kent, K. Seo.

NOTE: Available at https://tools.ietf.org/html/rfc2401.

[i.31] IETF RFC 6863: "Analysis of OSPF security according to the keying and authentication for routing protocols (karp) design guidelines", March 2013, S. Hartman, D. Zhang.

NOTE: Available at https://tools.ietf.org/html/rfc6863.

[i.32] IETF RFC 8205: "BGPsec protocol specification", September 2017, M. Lepinski, K. Sriram.

NOTE: Available at https://tools.ietf.org/html/rfc8205.

[i.33] IETF RFC 5246: "The Transport Layer Security Protocol, version 1.2", August 2008, T. Diersk, R. Escola.

NOTE: Available at https://tools.ietf.org/html/rfc5246.

[i.34] J. Small: "Patterns in network security: An analysis of architectural complexity in securing RINA networks", Boston University Computer Science Department, Master thesis, 2012.

NOTE: Available at https://open.bu.edu/handle/2144/17155.

[i.35] IETF draft-ietf-tsvwg-natsupp: "Stream Control Transmission Protocol network address translation", July 2017, R. Stewart, M. Tuexen, I. Rengeler.

[i.36] R. Lychev, S. Goldberg and M. Schapira: "BGP security in partial deployment: Is the juice worth the squeeze?", ACM SIGCOMM 2013.

[i.37] M. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang and M. Zhani: "Data center network virtualization: A survey", IEEE Communications Surveys and Tutorials, vol. 15, no. 2, 2013.

[i.38] B. Schneier: "A plea for simplicity: You can't secure what you don't understand", Information Security, 1999.

NOTE: Available at https://www.schneier.com/essays/archives/1999/11/a_plea_for_simplicit.html.

[i.39] J. Mirkovic and P. Reiher: "A taxonomy of DDoS attacks and DDoS defense mechanisms", ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pages 39-53, 2004.

[i.40] IEEE 802.1aq^{TM}: "Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 8: Shortest Path Bridging, 802.1aq", April 2012.

NOTE: Available at https://standards.ieee.org/standard/802_1aq-2012.html.

[i.41] E. Grasa, B. Gastón, S. van der Meer, M. Crotty and M. A. Puente: "Simplifying multi-layer Network Management with RINA", Proceedings of the TNC conference, 2016.

NOTE: Available at https://tnc16.geant.org/core/presentation/667.

[i.42] J. Day: "How Distributed is Distributed Management? Or can too many cooks spoil the broth?", Pouzin Society blog post.

NOTE: Available at http://pouzinsociety.org/node/55.

[i.43] PRISTINE Consortium: "D5.4 consolidated dif management system", PRISTINE deliverable D5.4, July 2016.

NOTE: Available at http://ict-pristine.eu/wp-content/uploads/2018/05/pristine-d54-consolidated-network-management-system_v1_0.pdf.

[i.44] ARCFIRE consortium, deliverable D3.1: "Integrated software ready for experiments: RINA stack, Management System and measurement framework", H2020 ARCFIRE, December 2016.

NOTE: Available at http://ict-arcfire.eu/wp-content/uploads/2017/10/arcfire_d31-final.pdf.

[i.45] V. Maffione: "Port of the dropbear ssh client/server for rina", December 2016.

NOTE: Available at https://github.com/vmaffione/rina-dropbear.

[i.46] M. Williams: "Prototype sockets emulator for rina", September 2017.

NOTE: Available at https://github.com/rlite/rina-dropbear.

[i.47] S. Vrijders, E. Trouva, J. Day, E. Grasa, D. Staessens, D. Colle, M. Pickavet and L. Chitkushev: "Unreliable inter process communication in Ethernet: migrating to RINA with the shim DIF," in 5th International Workshop on Reliable Networks Design and Modeling (RNDM-2013), 2013, pp. 97-102.

[i.48] IRATI Consortium: "IRATI Deliverable D2.4, third phase use cases, updated RINA specification and high-level software architecture", IRATI website, December 2014.

NOTE: Available at http://irati.eu/wp-content/uploads/2012/07/IRATI-D2.4-bundle.zip.

[i.49] IRATI Consortium: "Irati Deliverable D3.3, second phase integrated rina prototype for hypervisors for aunix-like OS", June 2014.

NOTE: Available at http://irati.eu/wp-content/uploads/2012/07/IRATI-D3.3-bundle.zip.

[i.50] V. Maffione: "Rina-tcp gateway implementation", September 2016.

NOTE: Available at https://github.com/rlite/rlite#71-rina-gw.

[i.51] S. Vrijders, V. Maffione, D. Staessens, F. Salvestrini, M. Biancani, E. Grasa, D. Colle, M. Pickavet, J. Barron, J. Day and L. Chitkushev: "Reducing the complexity of virtual machine networking", IEEE Communications Magazine, vol. 54, no. 4, pp. 152-158, April 2016.

[i.52] P. Teymoori, M. Welzl, S. Gjessing, E. Grasa, R. Riggio, K. Rausch and D. Siracussa: "Congestion control in the recursive internetwork architecture (RINA)", IEEE ICC 2016, Next Generation Networking and Internet Symposium, 2016.

[i.53] S. León, J. Perelló, D. Careglio, E. Grasa, D. Lopez and P. A. Aranda: "Benefits of programmable topological routing policies in rina-enabled large-scale datacentres", IEEE Globecom 2016, Next Generation Networks Symposium, December 2016.

[i.54] V. Maffione: "Prototype rina-enabled vrf implementation", May 2018.

NOTE: Available at https://github.com/rlite/rlite#72-iporinad.

[i.55] ARCFIRE consortium, deliverable D4.3: "Design of experimental scenarios; selection of metrics and kpis", H2020 ARCFIRE, January 2017.

NOTE: Available at http://ict-arcfire.eu.

[i.56] IEEE 802.11™: "IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Application Process (AP): instantiation of a program executing in a processing system intended to accomplish some purpose

NOTE: The Application Process definition aims to be quite abstract and applicable to a broad range of hardware and software (CPUs, FPGAs, ASICs and other platforms). See the definition of "Processing System" below.

Common Distributed Application Protocol (CDAP): application protocol component of a Distributed Application Facility (DAF) that can be used to construct arbitrary distributed applications, of which the DIF is an example

NOTE: CDAP enables distributed applications to exchange and operate structured data objects, rather than forcing applications to explicitly deal with serialization and input/output operations.

Data Transfer Control Protocol (DTCP): optional part of EFCP that provides the loosely-bound mechanisms

NOTE: Each DTCP instance is paired with a DTP instance to control the flow, based on its policies and the contents of the shared state vector.

Data Transfer Protocol (DTP): required data transfer part of EFCP consisting of tightly bound mechanisms found in all DIFs, roughly equivalent to IP and UDP

NOTE: When necessary DTP coordinates through a state vector with an instance of the Data Transfer Control Protocol. There is an instance of DTP for each flow.

Distributed Application Facility (DAF): collection of two or more cooperating Application Processes in one or more processing systems, which exchange information using the IPC services provided by a DIF and maintain shared state

NOTE: In some Distributed Applications, all members will be the same, i.e. a homogeneous DAF, or may be different, a heterogeneous DAF.

Distributed IPC Facility (DIF) layer: collection of two or more Application Processes cooperating to provide Interprocess Communication (IPC)

NOTE: A DIF is a DAF that does IPC. The DIF provides IPC services to Applications via a set of API primitives that are used to exchange information with the Application's peer.

Error and Flow Control Protocol (EFCP): data transfer protocol required to maintain an instance of a communication service within a DIF

NOTE: The functions of this protocol ensure reliability, order, and flow control as required. It consists of separate instances of DTP and optionally DTCP, which coordinate through a state vector.

flow: service provided by an EFCP-instance to an application process

NOTE: The binding between an EFCP-instance and the application process using it is called a port.

Flow Allocator (FA): layer management component of the IPC Process that responds to Allocation Requests from Application Processes

NOTE: A Flow Allocator Instance (FAI) is created for each Allocate Request. The FAI is responsible for:

1) finding the address of the IPC-Process with access to the requested destination-application;
2) determining whether the requesting Application Process has access to the requested Application Process;
3) selecting the policies to be used on the flow;
4) monitoring the flow; and
5) managing the flow for its duration.

Inter Process Communication (IPC): service provided by a DIF to two or more instances of Application Processes, allowing them to exchange information

IPC Process (IPCP): Application Process, which is a member of a DIF and implements locally the functionality to support and manage IPC using multiple sub-tasks

layer: set of protocol machines sharing state under a certain scope

NOTE: In the context of RINA, a layer is a Distributed IPC Facility.

(N)-DIF: nomenclature to indicate the rank of a DIF, as a basis to describe its relationship with DIFs in the ranks above ((N+1)-DIF)) and below ((N-1)-DIF)

peer IPCP: IPCP in the same DIF that is one hop away, without requiring another IPCP to act as a relay

NOTE: In general, peer IPCPs should have an N-1 DIF in common.

processing system: hardware and software capable of executing programs instantiated as Application Processes that can coordinate with the equivalent of a "test and set" instruction, i.e. the tasks can all atomically reference the same memory

Protocol-Data-Unit (PDU): string of octets exchanged among the Protocol Machines (PM)

NOTE: PDUs contain two parts. The PCI (Protocol Control Information), which is understood and interpreted by the DIF, and User-Data, that is incomprehensible to this PM and is passed to its user.

Protocol Machine (PM): implementation of the protocol logic that exchange state information with a peer PM by inserting protocol control information into a PDU on one side (sender), and striping it in the other side (receiver)

Relaying/Multiplexing-Task (RMT): RMT performs the real time scheduling of sending PDUs on the appropriate (N-1)-ports of the (N-1)-DIFs available to the RMT

NOTE: This task is an element of the data transfer function of a DIF. Logically, it sits between the EFCP and SDU Protection.

Resource Allocator (RA): component of the DIF that manages resource allocation and monitors the resources in the DIF by sharing information with other DIF IPC Processes and the performance of supporting DIFs

Resource Information Base (RIB): logical representation of the local repository of the objects exposing the externally visible state of an Application Process

NOTE: Each member of the DAF maintains a RIB. A Distributed Application may define a RIB to be its local representation of its view of the distributed application.

RIB Daemon: layer management component of the IPC Process that optimizes the requests for information from the other layer management tasks of the IPCP

NOTE: Each local Application Process participating in a Distributed Application may have several sub-tasks or threads. Each of these may have requirements for information from other participants in the distributed application on a periodic or event driven basis.

Service-Data-Unit (SDU): amount of data passed across the (N)-DIF interface to be transferred to the destination application process

NOTE: The integrity of an SDU is maintained by the (N)-DIF. An SDU may be fragmented or combined with other SDUs for sending as one or more PDUs.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ACL	Access Control List
ADDR	Address
AP	Application Process
API	Application Programming Interface
AS	Autonomous System
ASIC	Application-Specific Integrated Circuit
ASN	Abstract Syntax Notation
BGP	Border Gateway Protocol
BS	Base Station
BSS	Basic Service Set
CACEP	Common Application Connection Establishment Phase
CDAP	Common Distributed Application Protocol
CEPID	Connection EndPoint IDentifier
CMIP	Common Management Information Protocol
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSP	Communication Service Provider
DAF	Distributed Application Facility
DC	Data Centre
DCCP	Datagram Congestion Control Protocol
DDoS	Distributed Denial of Service
DIF	Distributed IPC Facility
DMM	Distributed Mobility Management
DNS	Domain Name System
DNSSEC	DNS Security
DSCP	Differential Services Code Point
DST	Destination
DTCP	Data Transfer Control Protocol
DTP	Data Transfer Protocol
ECN	Explicit Congestion Notification
EFCP	Error and Flow Control Protocol