Enhancement of Feedback Congestion Control Mechanisms by Deploying Active Congestion Control

Yoganandhini Janarthanan

Aug 30,2001

Committee : Dr.Gary Minden

Dr. Joseph Evans

Dr.Perry Alexander

Outline

- Introduction
  - Traditional Vs. Active Networks
  - Motivation
- Active Congestion Control Framework
  - Concept
  - Terminology
  - Components
- Specification and Verification
- Summary

Introduction

- Traditional networks - transfer bits from one end system to another, with minimum computation.
- Active networking - from passive carrier of bits to a more general computing engine.
- Nodes can perform computations on user data as it traverses the network.
- Users inject customized programs into the nodes, that modify/redirect/store user data flowing through the network.

Introduction

- Shortcoming of traditional networks: Difficulty to accommodate new services in the existing architectural model.
- Processing can be customized on a per user/per application basis as compared to the traditional routers, which send the user data opaquely.
- Congestion is a prime candidate for active networking, as it is an intra-network event and is potentially far removed from the application.

Motivation

- Congestion control makes a good case for active networking, enabling schemes that are not possible within the conventional view of the network.
- The sender-adaptation model presents a number of challenges:
  - Loss is the only mechanism for determining available bandwidth.
  - A time interval is required for the sender to detect congestion and adapt in order to bring losses under control. During this interval, the receiver experiences uncontrolled loss, resulting in the reduction in the quality of service.

Motivation

- These challenges overcome by moving the adaptations that the sender takes during congestion, into the network.
- ICMP source quench: Congestion control mechanism in the network layer. Deficiencies:

  How the router decides when to send the source quench message, how often to send, when to stop and how the host reacts to source quench message are unclear.
- The active congestion control scheme proposed in this thesis tackles these issues.

Active Congestion Control

- The endpoint congestion control algorithms moved into the network, where they can immediately react to congestion.
- The current state of the endpoint's feedback algorithm is included in every packet.
- This state information is used by the router during congestion.
- Active congestion control reduces the duration of each congestion event and is very effective in high-speed networks.

Terminology

- Active router

  Any router in the network, having the active functionality.

- Component

  An entity that implements a piece of functionality in the framework.

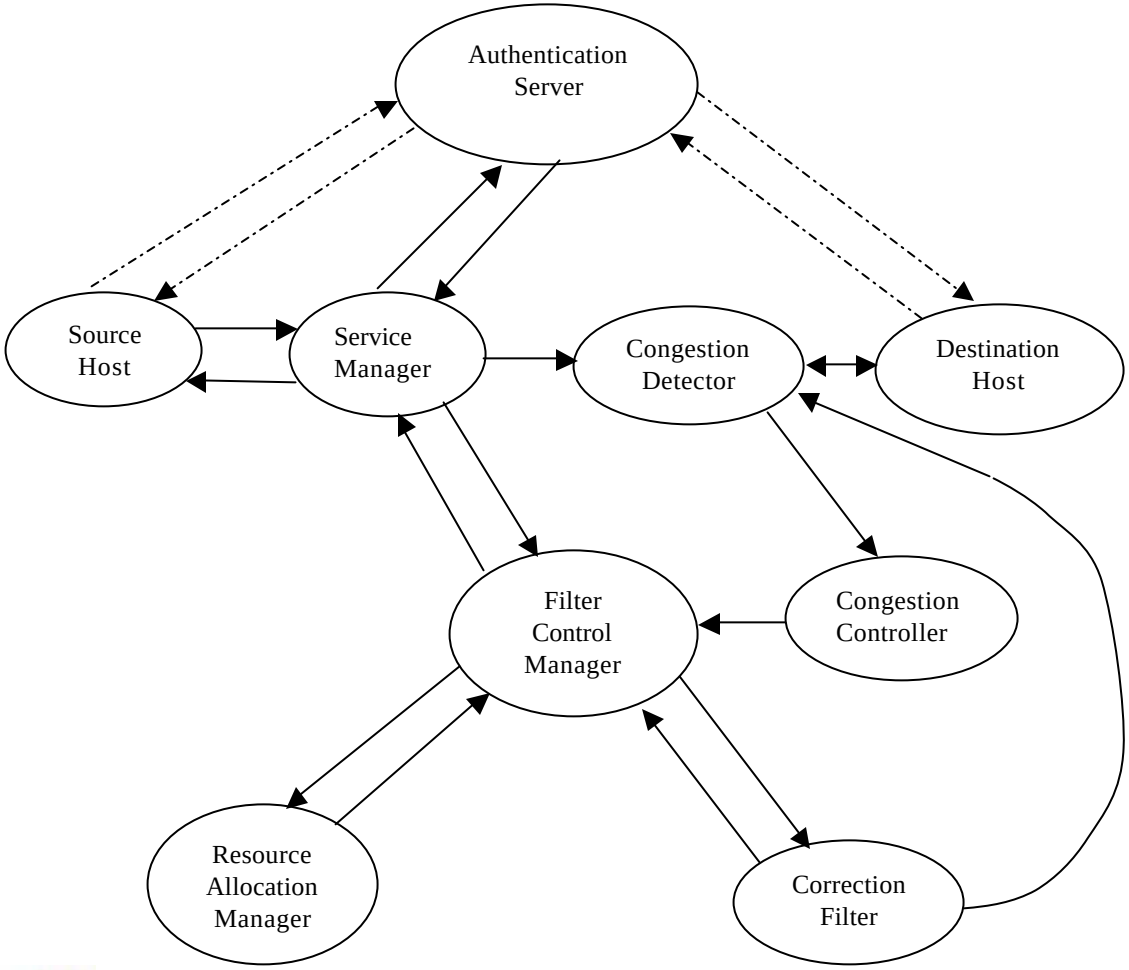
- Active host

  Any entity which interacts with, and uses the services provided by the active router. (Further classified into trusting and non-trusting)

- Service

  Functionality available to the network, through the use of one or more components

Components


Authentication Server
Source Host
Service Manager
Congestion Detector
Destination Host
Filter Control Manager
Congestion Controller
Resource Allocation Manager
Correction Filter


I T T C

Components

- Service manager
- Authentication server
- Filter control manager
- Resource allocation manager
- Congestion detector
- Congestion controller
- Correction filter

Components

- Service manager
  - Receives the message from the hosts
  - Sends the packets to the Authentication Server for authentication
  - If the result of authentication is a success,depending on the type of the packet,
    - It directs the Filter Control Manager to install/uninstall filters, after checking the priorities of the filter.
    - It directs the data packets to the Congestion Detector

Components

- Authentication server
  - Invoked by the Service Manager for verification of authenticity.
  - Also invoked by some hosts, which don't trust the Service Manager.
  - Three levels of authentication
    - Source Authentication
    - Message Integrity check
    - Source Reliability
  - Two types - public key cryptography based and secret key cryptography based authentication.

Components - Authentication Server

- Certification authority (CA)
- Key distribution center (KDC)

- Message integrity
  - In the public key method, hash of the message is computed and signed using the sender's private key.
  - In the secret key method, the message digest is signed using the shared secret key.

- Source reliability is checked by making sure that the sender hasn't misbehaved in the past.

Components

- Filter control manager
  - Contacts the Resource Allocation Manager, whenever it gets a filter install/uninstall/update request, to make sure that the required resources are available and to avoid exploitation by a single user.
  - Based on the availability of resources, it installs the filter specified by the sender in the active router.
  - In case no filter is specified, the default filter is used, which just drops packets during congestion.

Components

- Resource allocation manager
  - Checks if resources are available for the particular request.
  - Makes sure that a single host doesn’t hog the resources.
  - Maintains the information about the number of requests from each host, the memory used, the priority of each of the filters, etc.
  - Handles preemption of the allocated resources, depending on the priority of the new request. For this purpose, it uses the filter setup and filter holding priorities.

Components

- Congestion detector
  - The data packets arrive here, from the Service Manager.
  - Two congestion detection algorithms have been used:
    - Drop tail: congestion is determined by the *buffer size*. Once the buffer is full, the router starts dropping packets.
    - RED (Random Early Detection): packets are marked before the queue is full.
    - The probability that an arriving packet is marked for discard is proportional to the amount that the router's current queue length exceeds a threshold.

Components

- Congestion detector (continued)
  - Two separate algorithms in RED; algorithm for computing the average queue size and the algorithm for calculating the packet-marking probability.
  - The congestion detector sends the packets to either the host or the congestion controller, depending on whether the packets are marked or not.

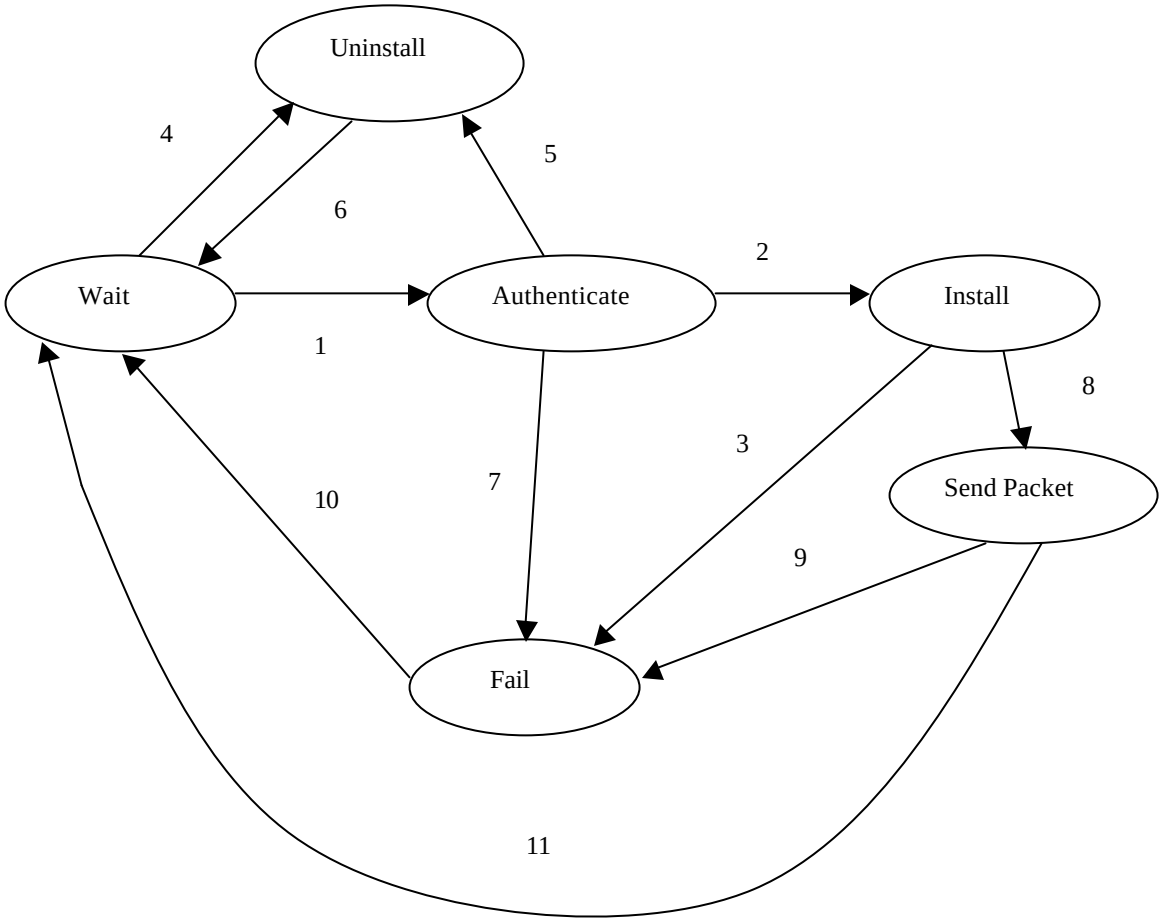
Components

- Congestion controller
  - Forwards the packet to the correction filter, through the filter control manager.
  - Sends a packet with the new window size to the source.
- Correction filter
  - The user specifies the filter to be used during congestion.
  - The rudimentary default filter deletes all packets.
  - A more sophisticated filter does some sort of traffic editing - e.G., Dropping the P and B frames (frames with less priority/information) in MPEG, during congestion.

Finite State Machine Model

- The model is decomposed into a set of states and the working of the model is embedded in the transitions between the states.
- The correctness of the model is verified by validating that for any set of valid inputs to the machine, the Finite State Machine Model generates correct outputs and/or proceeds to valid termination states.

Finite State Machine of Service Manager



States in the Service Manager Model

State	Explanation
Wait	The Service Manager is waiting for incoming packets
Authenticate	The Service Manager sends a message that needs to be authenticated to the Authentication Server.
Install	The Service Manager directs the Filter Control Manager to install the filter, by providing the required information.
Uninstall	The Service Manager directs the Filter Control Manager to uninstall the filter. This may either be due to a timeout or an explicit request from a sender regarding changing the filter.
Send	Packets are sent towards the destination.
Fail	Failure occurs in various states, namely, authenticate, install or due to corrupted packets.

Events in the Service Manager Model

Transition Event	Explanation
1	The Service Manager receives a message.
2	The authentication of the message is successful.
3	The filter service is preempted because of the request for resources with a higher priority.
4	There is a timeout at the Service Manager, due to inactivity in a particular process.
5	An explicit request is made by a sender to change the type of filter used, and this request is authenticated.
6	The filter is uninstalled, and the resources associated with it are released.
7	Authentication of a message failed.
8	Successful installation of the filter followed by subsequent packets being sent by the Service Manager.
9	Packets not sent due to corrupted bits.
10	Packet was not sent (Failure) and the Service Manager waits for the next request.
11	Packets sent successfully and the Service Manager waits for the next request.

Features of Our Model

- Dynamism
- Minimum Oscillation
- Convergence
- Robustness
- Compatibility

Specification And Verification

- Specification: Process of describing a system and its properties. Formal specification uses a language with mathematically defined syntax and semantics.
- Verification: Process of mathematically proving the veracity of the specification.
- Specification and verification increase confidence in the system by revealing inconsistencies, ambiguities and incompleteness.

Specification and Verification

- Specification and verification done using SPIN/Promela.
- Used Xspin, the graphical interface for SPIN.
- Verified the logical consistency and the correctness of the model.

Specification

- Message parameters
  - Packet type
  - Process ID
  - Authentication info
  - Sequence numbers
  - State information
  - Filter setup priority
  - Filter holding priority
  - Source/destination address
  - Failure information
  - Miscellaneous attributes

Specification

- TLV encoding
- Error conditions
  - Resource unavailable
  - Authentication failure
  - Filter installation failure
  - Errors due to corrupted packets
  - Timeout errors
- System model

Specification

- Message types

```
mtype = {   snd_data,           //send data
            Snd_ack,            // send acknowledgement
            Auth_req,           // authorization request
            Auth_rep,           // authorization reply
            Filter_install_req, // filter installation request
            Filter_install_rep, // filter installation reply
            Res_alloc_req,      // resource allocation request
            Res_alloc_rep,      // resource allocation reply
        }
```

Specification

- Packet types

  User-defined data types are supported through typedef definitions.

```
typedef filter_install_pkt {
    mtype msgtype;
    byte flt_id;
    byte src_id;
    byte auth_info;
    byte setup_prio;
    byte hold_prio;
    byte state_info;
    byte result;
} ;
```

Specification

- Non-interleaved execution

  Atomic statements

- Temporal claims

```
never {
    do
    ::  Authent_failure->break;
    :: skip;
    od;
    do
    :: Flt_installed;
    od;
}
```

Verification

- Correctness and completeness of composition
  - Assertion violations
  - Unreachable code
  - Absence of deadlocks
  - Absence of livelocks
  - Temporal properties

Verification Scenarios

- Increasing the number of hosts and routers
  - Number of hosts increased to 3.
  - Number of routers increased to 3.
- Interleaving trusting and non-trusting hosts
  - Trusting host sends data to non-trusting host.
- Changing the filter
  - Used the rudimentary filter which dropped all the packets, and the priority based filter, which dropped packets with least priority.
- Active and non-active hosts
- Active And non-active routers

Conclusions

- The State Space and memory used for verification increases as the number of components in the system increases, as the complexity of the system increases.
- In the absence of active hosts, the framework functioned normally, thereby exhibiting compatibility.
- The framework has functionally separate components, hence changing the filters was easy.

Summary

- A framework for active congestion control has been proposed.
- The components in the framework are identified. The interaction among the various components are observed using the finite state machine model.
- The specification and verification of the proposed framework has been executed under different verification scenarios.

Thank You!