

# Consistent Updates for Software-Defined Networks: Change You Can Believe In!

**Mark Reitblatt**, Nate Foster,  
Jen Rexford, and Dave Walker





“[A] network change was performed as part of our **normal** AWS scaling activities... This change **disconnected** both the primary and secondary network simultaneously, leaving the affected nodes completely isolated from one another.”

The New York Times Technology | Personal Tech | Business Day

## Bits

### Amazon Cloud Failure Takes Down Web Sites

By CLARE CAIN MILLER | April 21, 2011, 4:40 PM

10:28 a.m. | Updated to reflect status of the problem on Friday.

A widespread failure in Amazon's Web services business was still affecting many Internet sites on Friday morning, highlighting the risks involved when companies rely on so-called cloud computing.

The problem, which began early Thursday morning, affected sites including Quora.com, Reddit.com, GroupMe.com and Sevigr.com, which all posted messages to their visitors about the issue. Most of the sites have been inaccessible for hours, and others were only partly operational.

The Web companies use Amazon's cloud-based service to serve their Web sites, applications and files. Amazon's customers include start-ups like the social networking site Foursquare but also big companies like Pfizer and Nasdaq.

Amazon, which is a leader in this business, lets these companies rent space on its servers and take advantage of its big data centers and computing power. But that gives the companies little control if the servers fail.

"We don't think the cloud is enterprise-ready," said Jimmy Tam, general manager of Peer Software, which provides data backup for businesses. "Are you really going to trust your corporate jewels to these cloud providers?"

# Prior Work

**Seamless Network-Wide IGP Migrations\***

Laurent Vanbergher, Stefano Vissicchio,  
Cristel Paísseau, Pierre François, Olivier Bonaventure

<sup>1</sup>Université catholique de Louvain<sup>1</sup>, Roma, The University<sup>2</sup> Internet Initiative Japan<sup>3</sup>  
(laurent.vanbergher, pierre.francois, olivier.bonaventure)@uclouvain.be  
vissicchio@ida.uniroma3.it, cristel@ijlab.jp

**ABSTRACT**

Network-wide migration of a routing protocol, such as the replacement of a routing protocol by the modification of its implementation, may require the performance, scalability, maintainability, and security of the entire network. However, such migration are an important source of errors for network operators as the implementation complexity, the need to bring and service-affected outage.

In this paper, we propose a methodology which addresses the problem of seamlessly migrating the configuration of multiple distributed link-state routing protocols. We illustrate the benefits of our methodology by considering the general migration operation, including the addition or the removal of specific operators, including the addition or the

removal of part of routes within the network of a large network operator (ISP). Many other specific protocols (BGP, LDP or PGM, etc) may be used in a similar way. As the network operator is able to use our tool to do the migration, it can be used to migrate the network-wide IGP configuration [1]. Migrating an IGP is a complex problem since all of the routers have to be reconfigured in a proper manner. Single routers like migrating the network with the new configuration do not work since most of the routers have many traffic [2-4]. Therefore, IGP migration has to be performed gradually, while the network is still running and available to the users. Unfortunately, network operators have to migrate the network-wide IGP configuration in a manual way.

## Seamless IGP migration

**Characterization and Analysis of Network Operators (CANA) (Workshop on Characterization, Analysis, and Network Operators)**

General Network Algorithms, Measurement, Security, Scalability, Service Quality, Traffic Engineering, Traffic Engineering, Traffic Engineering, Traffic Engineering

**1. INTRODUCTION**

Aspects of network routing protocols, including Internet Protocol (IP) and Open Shortest Path First (OSPF), are well known. However, most of the research on routing protocols is still in the early stages of development.

**2. RELATED WORK**

Previous work on routing protocols, including Internet Protocol (IP) and Open Shortest Path First (OSPF), are well known. However, most of the research on routing protocols is still in the early stages of development.

**Avoiding transient loops during the convergence of link-state routing protocols**

Pierre François and Olivier Bonaventure  
Université catholique de Louvain

**Abstract.** When using link-state protocols such as OSPF or IS-IS, transient loops may occur transiently when the network converges. In this paper, we present a procedure that lets the network converge to a loop-free forwarding state. The procedure is based on the use of a special forwarding state. The procedure is based on the use of a special forwarding state. The procedure is based on the use of a special forwarding state.

## Avoiding transient loops

**Internet Service Providers, link convergence in case of failures in a key problem that must be solved [1]. Many operators are migrating to OSPF, including factors and procedures to avoid all packet losses.**

Various are actively working on improving their implementation to achieve faster convergence [1]. However, the fast convergence procedure is complex as it involves detecting the failure on the affected router, propagating a new Link State Packet (LSP) describing the failure, flooding this new LSP and finally updating the Forwarding Information Base (FIB) in all of the routers using the failed resources in the network. Subsequent convergence has been made possible, but the sub-50 msec target can only be achieved by the means of a local restoration scheme. Achieving very fast convergence in an IGP network will require innovative methods to quickly restore the network. In this paper, we propose a methodology to avoid transient loops during the convergence of link-state routing protocols. We illustrate the benefits of our methodology by considering the general migration operation, including the addition or the removal of specific operators, including the addition or the

**Consensus Routing: The Internet as a Distributed System**

John P. John<sup>1</sup>, Ethan Katz-Bassett<sup>2</sup>, Arvind Krishnamurthy<sup>3</sup>, Thomas Anderson<sup>4</sup>, Arun Venkatesan<sup>5</sup>

**Abstract**

Internet routing, especially transitive routing, has traditionally been implemented in a centralized manner. However, network operators are migrating to distributed routing protocols such as OSPF, which require a distributed manner. This paper presents a distributed routing protocol, called Consensus Routing, which is designed to be implemented in a distributed manner. Consensus Routing is designed to be implemented in a distributed manner. Consensus Routing is designed to be implemented in a distributed manner.

## Consensus routing

Internet routing, especially transitive routing, has traditionally been implemented in a centralized manner. However, network operators are migrating to distributed routing protocols such as OSPF, which require a distributed manner. This paper presents a distributed routing protocol, called Consensus Routing, which is designed to be implemented in a distributed manner. Consensus Routing is designed to be implemented in a distributed manner. Consensus Routing is designed to be implemented in a distributed manner.

**Graceful Network State Migrations**

Saigal Basu, Member IEEE, Yuesha Zhu, and Chen-Nee Chah, Senior Member IEEE

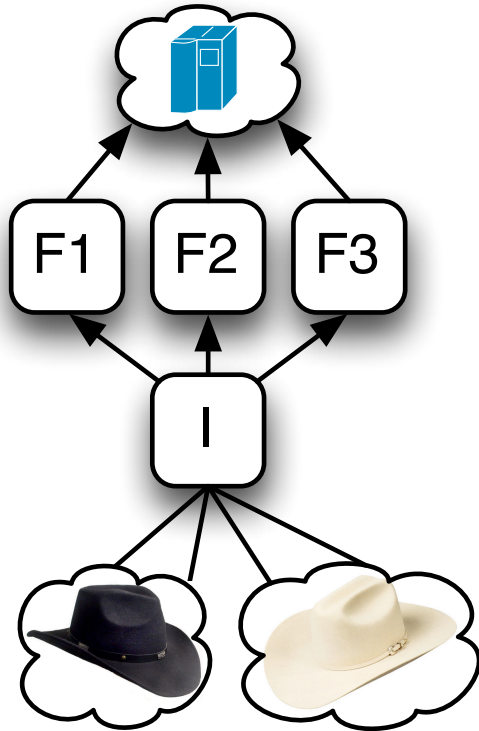
**Abstract.** A significant challenge for network operators is to migrate network state during the migration of a network operator. This paper presents a methodology to avoid transient loops during the convergence of link-state routing protocols. We illustrate the benefits of our methodology by considering the general migration operation, including the addition or the removal of specific operators, including the addition or the

## Graceful state migration




**Internet Service Providers, link convergence in case of failures in a key problem that must be solved [1]. Many operators are migrating to OSPF, including factors and procedures to avoid all packet losses.**

Various are actively working on improving their implementation to achieve faster convergence [1]. However, the fast convergence procedure is complex as it involves detecting the failure on the affected router, propagating a new Link State Packet (LSP) describing the failure, flooding this new LSP and finally updating the Forwarding Information Base (FIB) in all of the routers using the failed resources in the network. Subsequent convergence has been made possible, but the sub-50 msec target can only be achieved by the means of a local restoration scheme. Achieving very fast convergence in an IGP network will require innovative methods to quickly restore the network. In this paper, we propose a methodology to avoid transient loops during the convergence of link-state routing protocols. We illustrate the benefits of our methodology by considering the general migration operation, including the addition or the removal of specific operators, including the addition or the

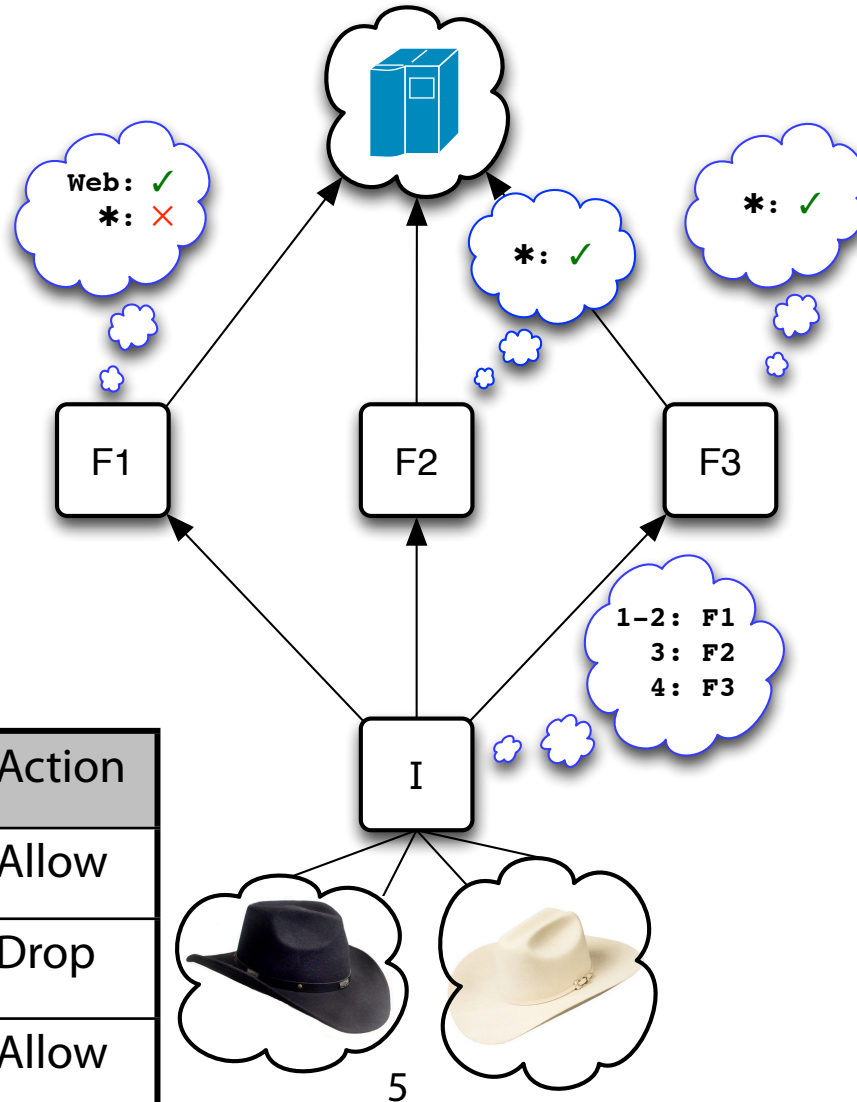
# Example






## Security Policy

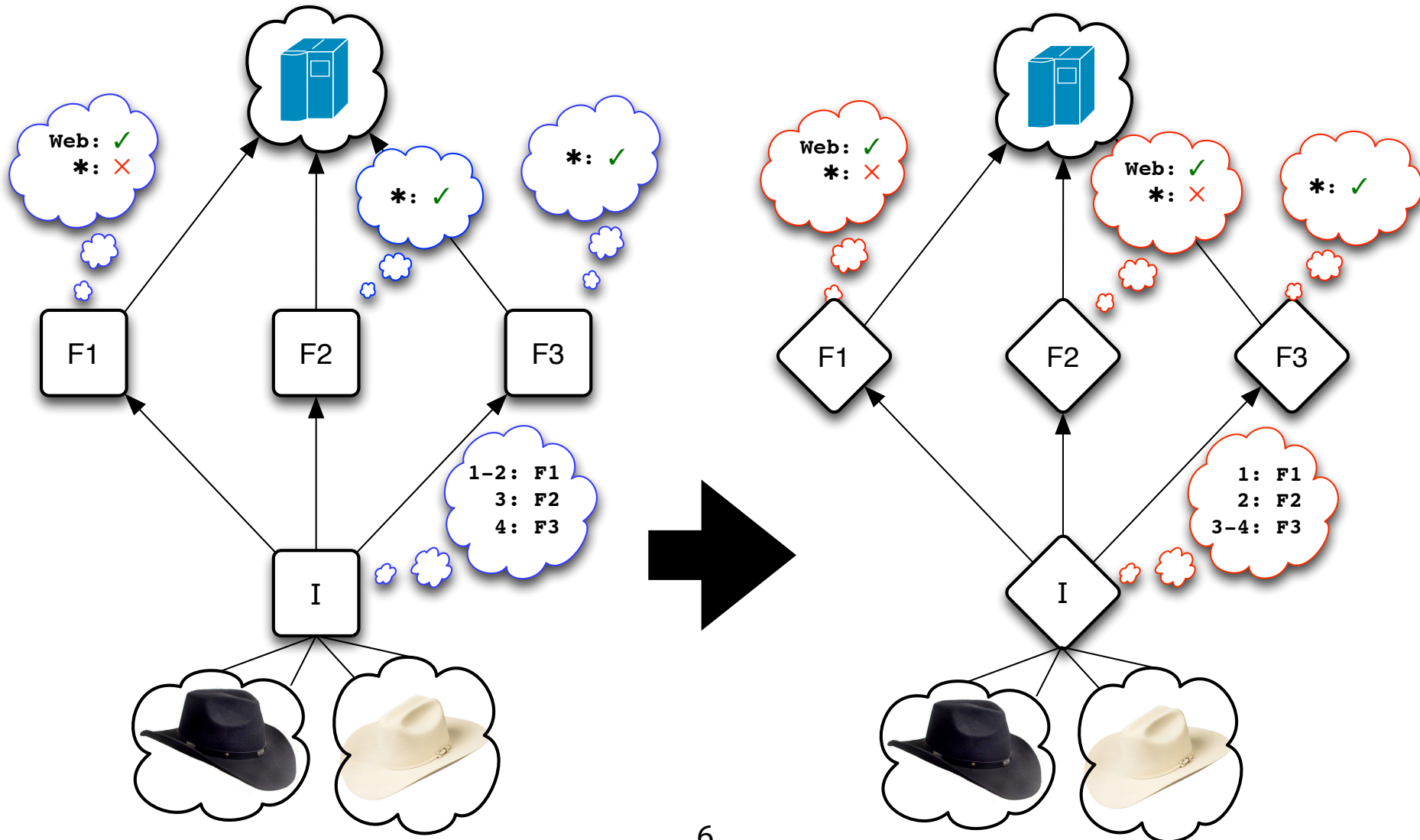
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow

# Initial Configuration

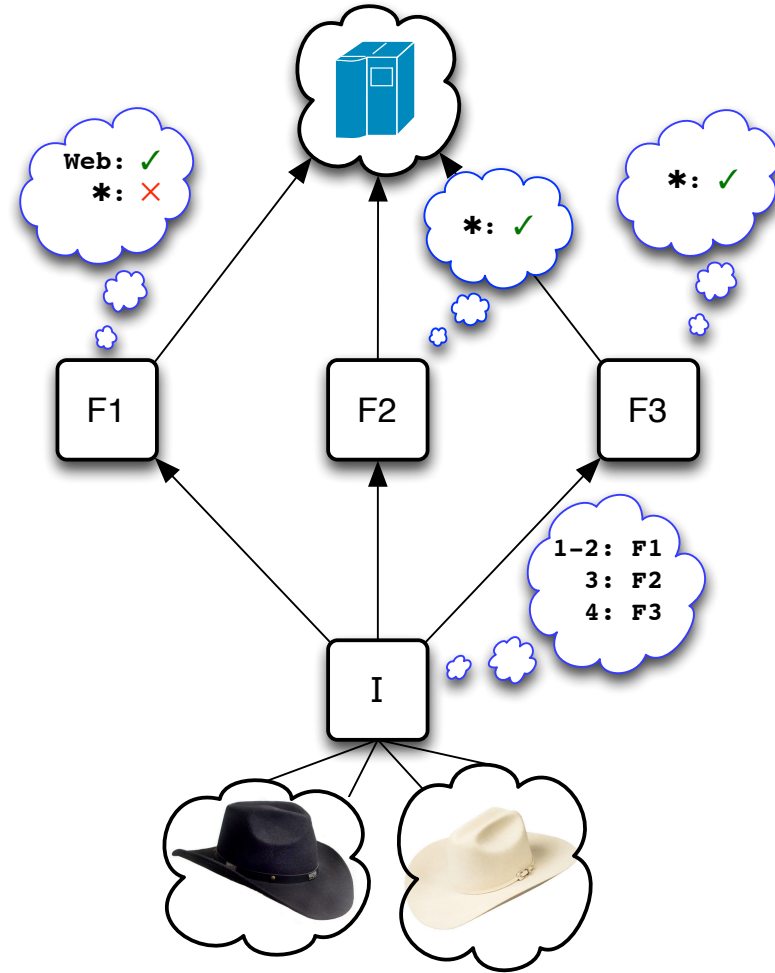
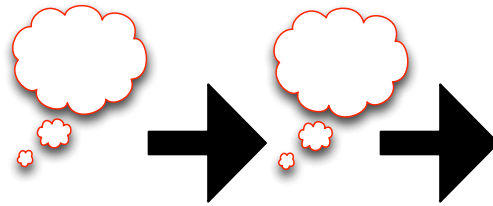
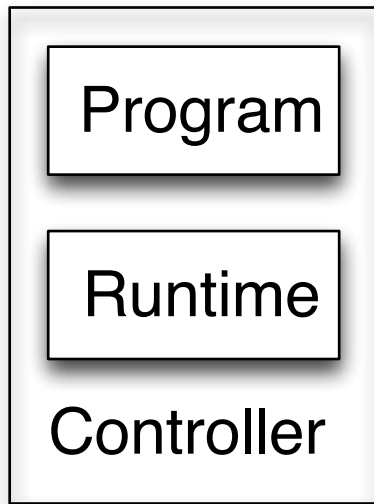


Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow

# Redistribute Configuration



# Software Defined Networks (SDN)

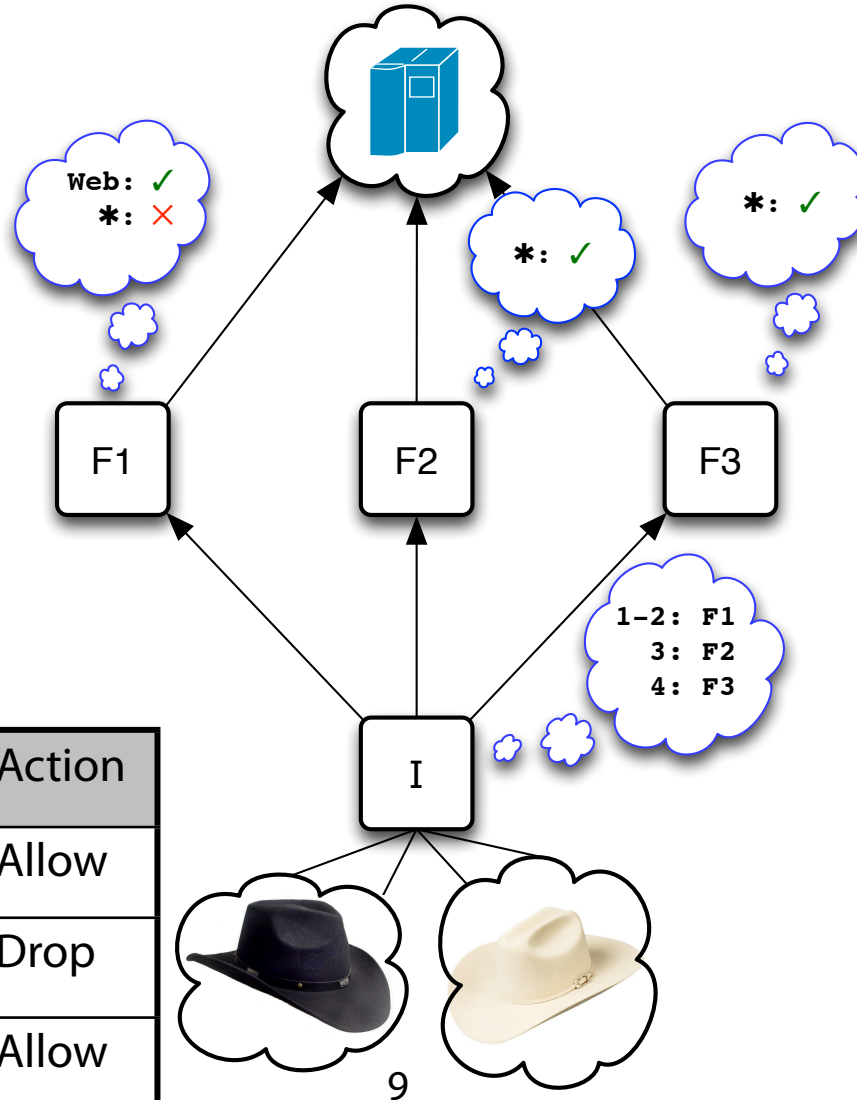





# SDN Program

```
void main() {  
    ... monitor ...  
    Conf = balance_load();  
    install(F1, Conf[F1]);  
    install( I, Conf[I] ) ;  
  
    ...  
}
```

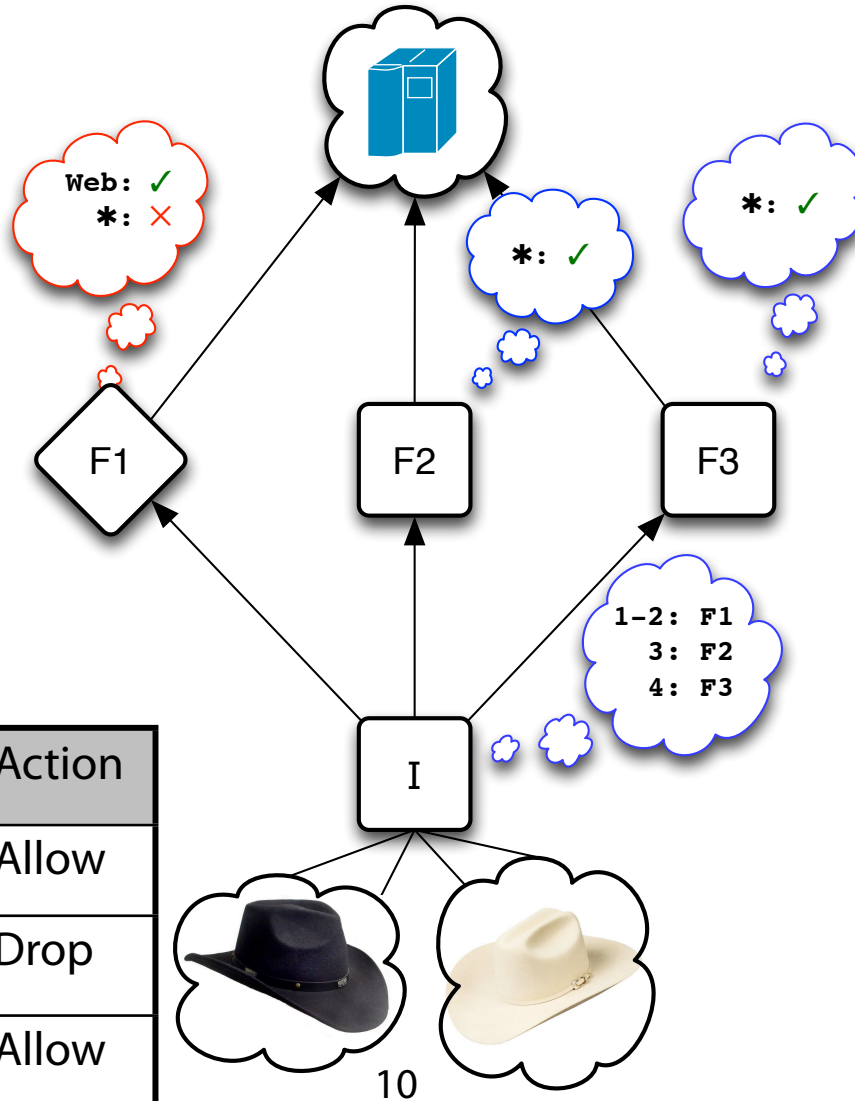





# Initial Configuration



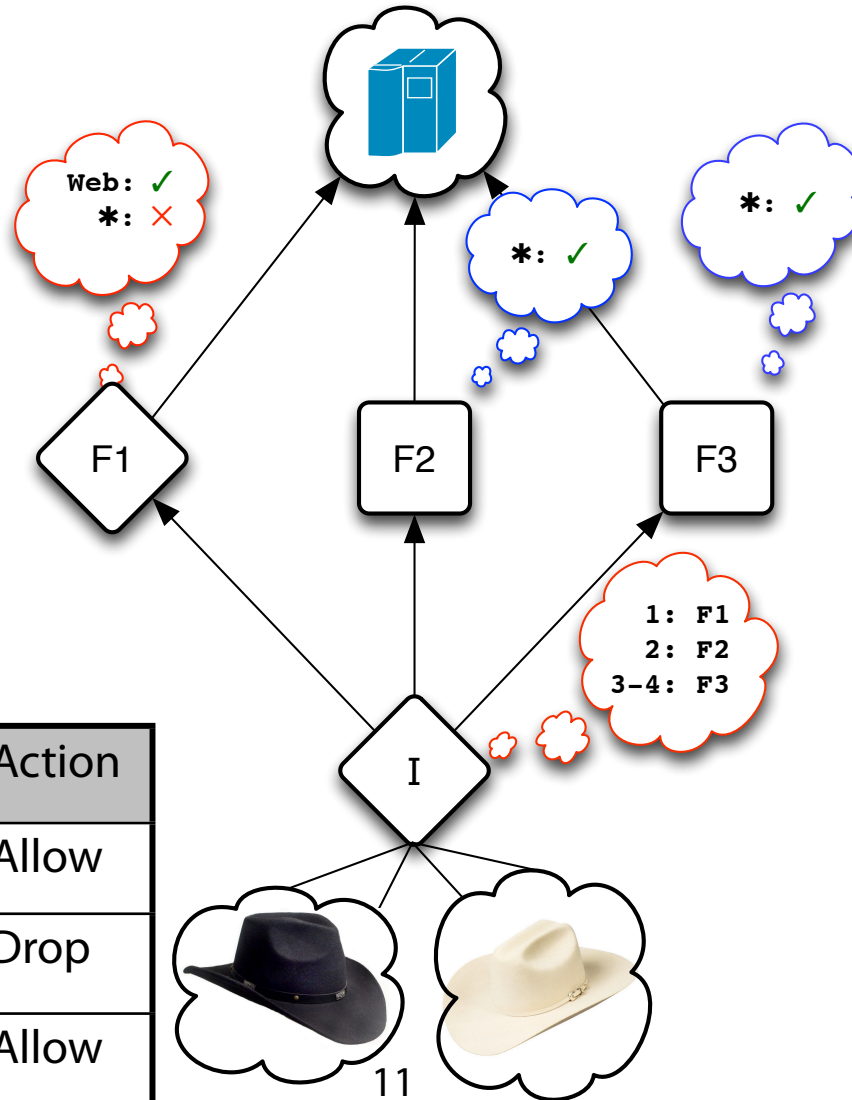
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Initial Configuration



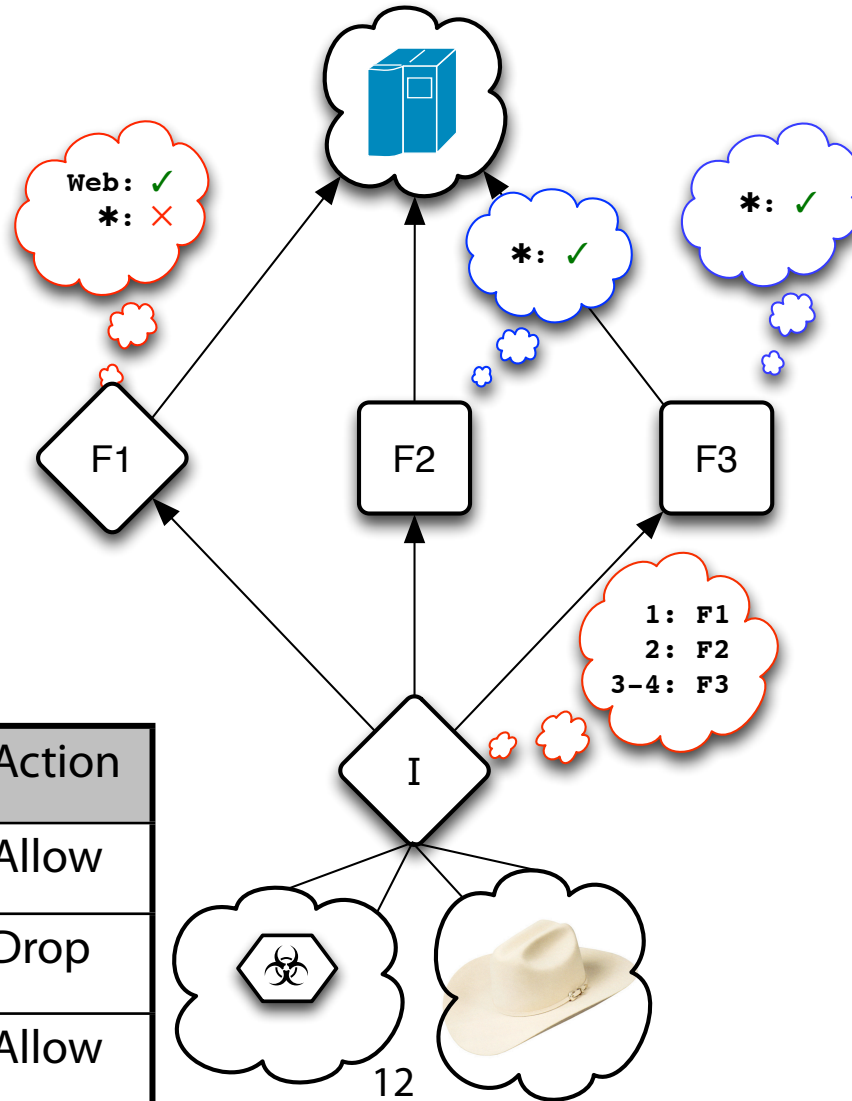
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Updating Configuration



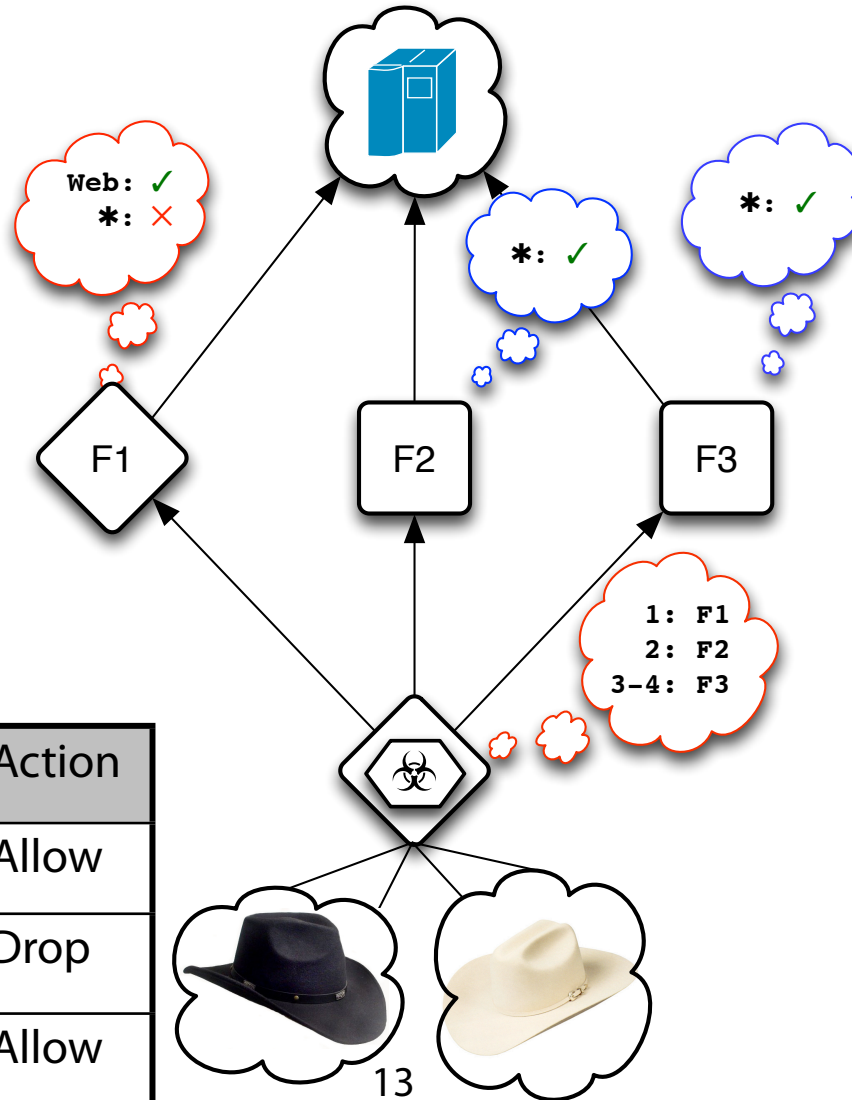
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Updating Configuration



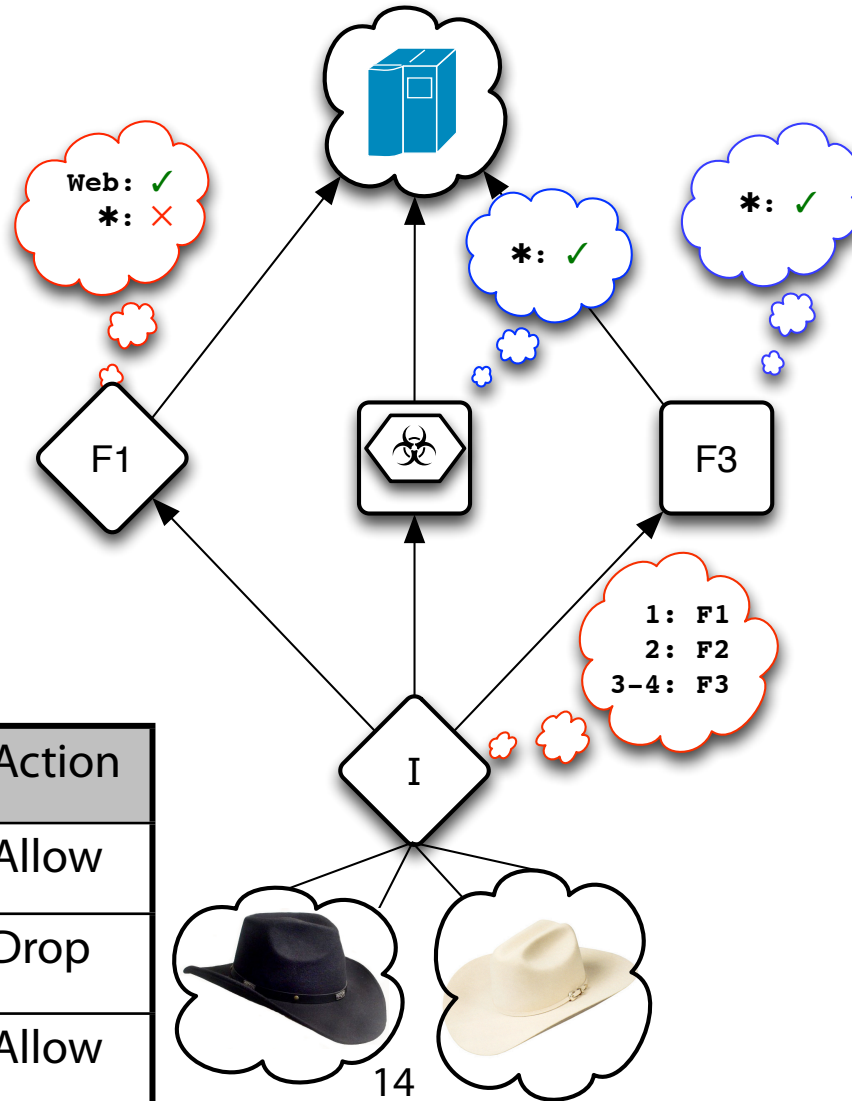
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Updating Configuration



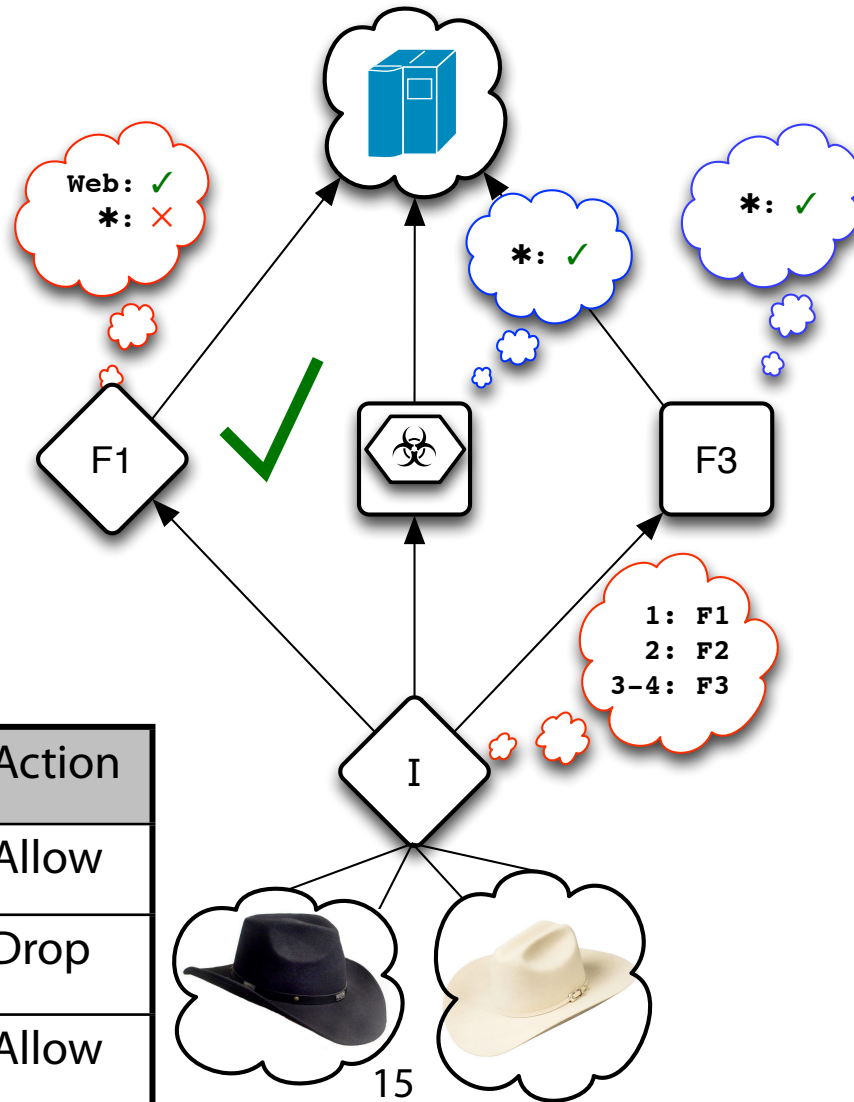
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Updating Configuration



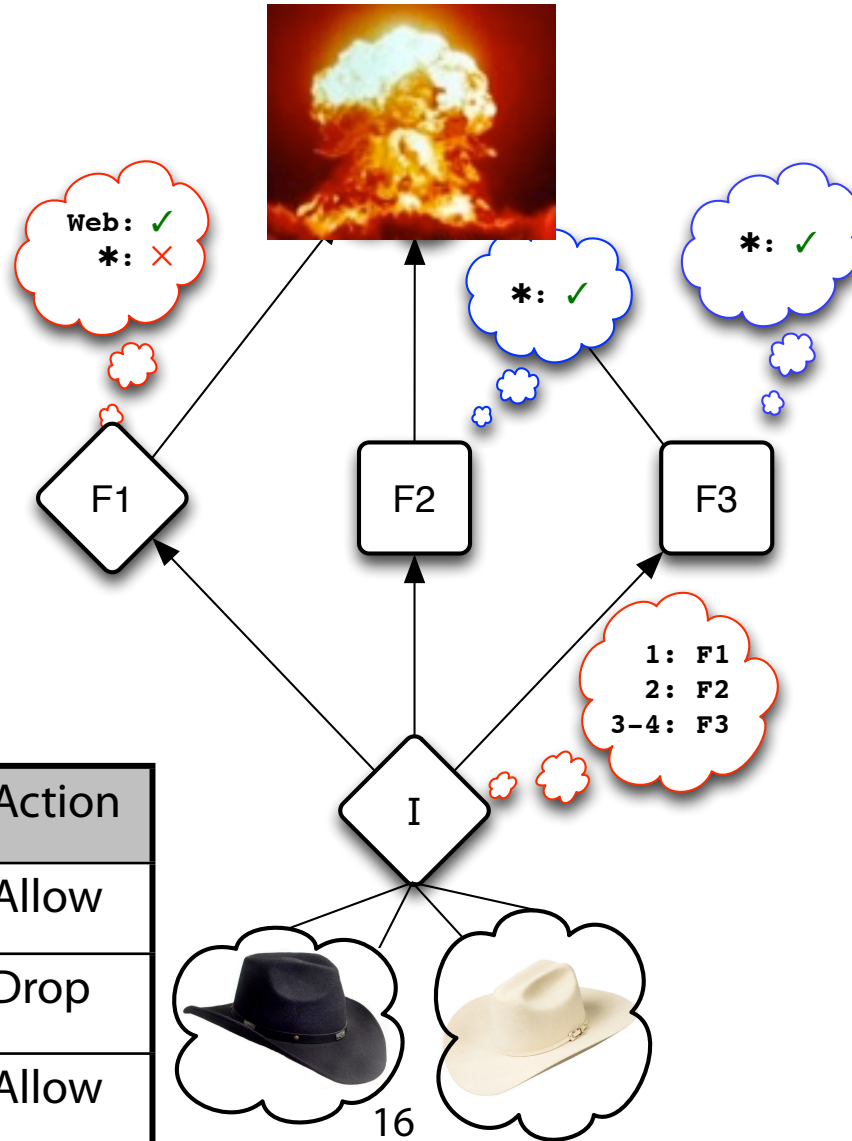
Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow




# Updating Configuration



Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow

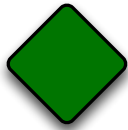
# Updating Configuration



Src	Traffic	Action
	Web	Allow
	Other	Drop
	Any	Allow



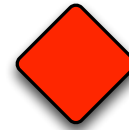
# Bad Update Order



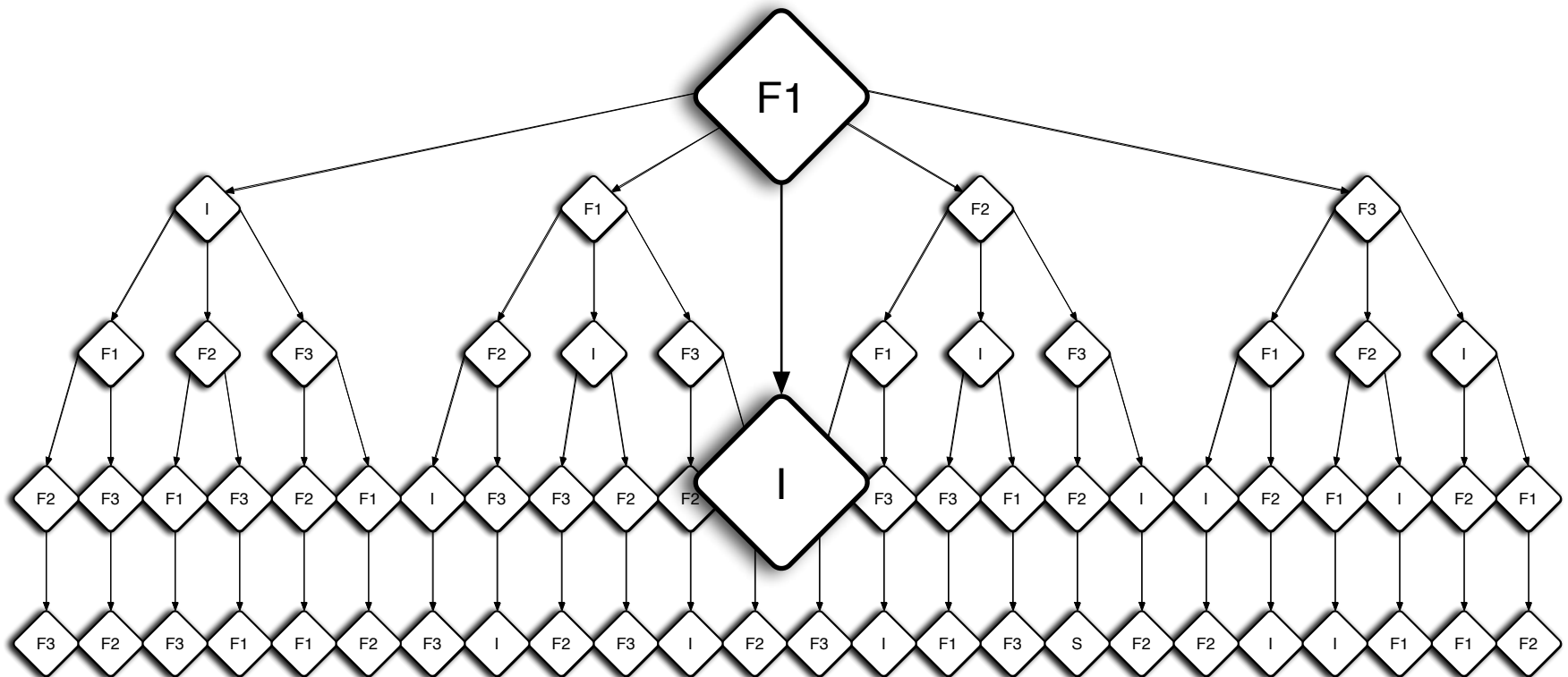
Safe



Broken Connectivity



Broken Security



# Towards a Solution

---

Updating individual switches doesn't work!

What's the solution?

- Don't implement updates rule-by-rule and switch-by-switch!
- Leverage the run-time system to handle tedious, low-level details

# SDN Program

---

```
void main() {  
    ... monitor ...  
    Conf = balance_load();  
    install(Conf);  
}
```

# Per-packet Consistency

*An update from configuration A to configuration B is **per-packet consistent** if each packet is routed according to either configuration A or B.*

# Path Properties

*A **path property**  $\phi$  specifies the legal paths that a packet can take through a network **N**.*

*Formally:  $\phi \subseteq \mathbf{Packet} \times \mathbf{Paths(N)}$ .*

- Loop-freedom
- “Block SSH traffic from 10/8”
- “All Web traffic waypoints through switch 5”

# SDN Program

---

```
void main() {  
    ... monitor ...  
    Conf = balance_load();  
    install(perpacket, Conf);  
}
```

# Beyond Path Properties

Not path properties:

- In-order delivery
- Flow affinity

*An update from configuration A to configuration B is **per-flow consistent** if each packet **in the same flow** is routed according to either configuration A or B.*

See paper for details

# 2-Phase Implementation

---

1. Instrument new configuration with version
2. Install instrumented configuration, leaving all old ingress rules active
3. Activate new ingress rules
4. Wait for old version packets to leave
5. Uninstall old configuration



# Future Work

## Implementation

- Naive prototype running
- Exploring performance optimizations

## Unplanned Change

- Highly responsive
- Weaker consistency

## Formal Verification

- Specification language for path properties
- Configuration verifier

# Ongoing Work



- **This paper**  
Network write abstraction
- **PRESTO '10, ICFP '11**  
Network read abstraction
- **POPL '12**  
Rich policy abstraction

# Questions?

---

Thank You



<http://frenetic-lang.org>

# Database Analogy

Network	Database
Fully routed packet	Read Transaction
Single hop routed packet	Read
Network update	Write Transaction
Single switch update	Write
Per-Packet Consistency	Snapshot Isolation