Traffic Congestion Patterns

6/28/1999

Robert Hanmer	Michael Wu
Lucent Technologies	Lucent Technologies
hanmer@lucent.com	mwu@lucent.com
2000 North Naperville Road	2000 North Naperville Road
PO Box 3033	PO Box 3033
Naperville, IL 60566-7033	Naperville, IL 60566-7033
voice: 630 979-4786	voice: 630 979-5615

Abstract

An examination of the successful techniques that have allowed earlier networks and network elements to be successful must be done to understand how to evolve networks. The patterns presented here describe techniques implemented in many Lucent Technologies switching systems to deal with traffic congestion issues. Handling congestion is one of the key roles of a network management system.

Congestion is the term used by telecommunications system designers to describe the situation when the call load exceeds the available resources of the system.

These patterns discuss ways that these systems deal with congestion. The approaches can be generally classified as protective or expansive controls. Protective controls reduce the amount of work by restricting access to resources. This protects the system from too much work. Expansive controls allow the system to use additional resources not available to call processing under normal conditions. This expands the possible actions that the system may take.

Introduction

> “Congestion occurs when the call load exceeds the capacity of available routes or trunks, or of common control equipment (for example, call registers that temporarily store call data while the call is being processed.) Congestion can also occur when the call load cannot be handled in available real time.” [MG]

Telecommunications system designers have studied the problem of system resource congestion for years. These same problems are evident in the distributed and client server computing systems of today. This article describes solutions that have worked in telecommunication systems over time.

Network management functions within telephone switching systems are designed to maintain a high level of service during periods of unusual traffic on systems that are engineered for less than full capacity. During these periods, machine performance naturally degrades due to resource contention and network delays. Even when normal traffic levels are restored the system continues to struggle to achieve normal performance levels. [GHHJ]

Internal congestion arises from two different categories: peripheral capacity and processor capacity. Common control or peripheral equipment capacity is related to how a system is engineered and how requests for service arrive. In the telecommunications world these requests for service are “traffic” and consist of telephone calls being handed off from one system to another as they progress throughout a network. Processor capacity (i.e. processor real time) is taken as the key driver behind a system’s "overload" capabilities. Peripheral capacity issues will be discussed here. For solutions to these problems refer to "A Pattern Language for Improving the Capacity of Reactive Systems" by Gerard Meszaros [Mesz].

We begin by discussing "patterns", the form in which the solutions we present are described. Then the context of telecommunications traffic and how it appears to switching systems is discussed. This is followed by the actual traffic congestion patterns.

Background

Patterns

This section lays the foundation that is required to understand the patterns presented in the remainder of this paper. A pattern is a description of a good, working solution to a problem that occurs again and again. A pattern contains a statement of the problem, the context where the problem exists, and a consideration of the various trade-offs involved in the solution, as well as the actual solution. Patterns only capture solutions that have stood the test of time and can be used whenever the problem occurs again. Generally patterns must be customized to a system's unique characteristics to be used.

What is contained in this article are solutions to problems related to peripheral control equipment congestion. These patterns describe functionality needed by a switching system to concentrate upon profitable work by avoiding unnecessary work and to decide in real time how calls are routed in a network. They each solve smaller parts of the larger problem of dealing with congestion, however when taken together they form the nucleus of an effective network management system.

The names of patterns appear as *underlined italics*.

Telephone Switching Systems and Traffic Congestion

A telephone switching system is a system that is intended to connect customer telephone circuits when the customers desire to communicate. These connections are created as a call is being established and are released when the customers hang up. Route selection and the selection of idle bandwidth are routinely done by these systems without human intervention.

Telephone switching systems are real-time, fault tolerant systems that attempt to maintain themselves without requiring constant human supervision (refer to *Minimize Human Intervention* [ACGH+]). But the systems generally consider that sometimes *People do know best* [ACGH+], so provisions for human oversight and altering normal actions are built into the system.

The patterns are primarily drawn from experiences with the 4ESS™ Switch, a high-capacity toll and tandem switch (refer to the *Hierarchical Routing* pattern below for an explanation of these terms). Traditionally the connections between switching offices are referred to as "trunks". The term "line" refers to the connections between switching offices and individual telephone sets. This pattern language only contains patterns related to congestion on trunks. Generally at the trunk level there are not restrictions that a particular call use a particular route, unlike when a telephone being called resides at the end of only one specific line.

Typically a telephone network requires a large number of switching systems, since individual systems have not been built that have the capacity required to concentrate all of the telephone traffic. Each of these systems is a node in the telephone network. Between these switching systems there are a large number of possible routes.

A large network in which every node is connected to every other node would be infeasible. Every route requires ongoing maintenance of the physical medium as well as maintenance within each switching system node that it connects to. But with more routes in the network, any specific telephone call has fewer switches to traverse; resulting in higher quality and less time required to complete the call.

A balance between number of routes with their associated expenses and the quality of service that is achievable and desirable in the network must be struck. The impact on traffic congestion should be obvious. If every switch is connected to every other with trunks in sufficient quantity then congestion will be non-existent. With fewer links, the possibility of routes concentrating on individual switches increases as well as the probability that that a switch will be subject to control equipment congestion.

Typically these networks have evolved with time. New switching systems replace older, smaller ones. Switching systems are quite expensive, so they are used as long as possible before this replacement occurs. Over time networks become very heterogeneous. This complicates any strategy for dealing with congestion since the switches may not all have the same abilities to control congestion.

The evolution of the network also requires that the systems be designed to be flexible in their routing. The various trunks that are installed in a new switching office might not be the same trunks as when the office is retired or replaced at the end of its useful life.

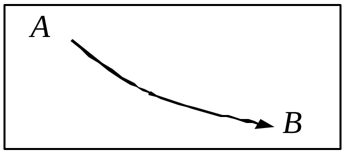
When a telephone network is being engineered usual telephone usage patterns are considered. Since switching equipment is expensive the systems are engineered to take advantage of the fact that each individual telephone line has much idle time. Most customers do not make continuous telephone calls. The systems are designed to allow expensive peripheral equipment to be shared among a number of lines or trunks. This common control equipment can be allocated as needed, and since most lines or trunks are not in use continually this sharing is feasible without diminishing the quality of service. But, shared resources can be needed by too many calls at the same time, resulting in peripheral capacity limitations.

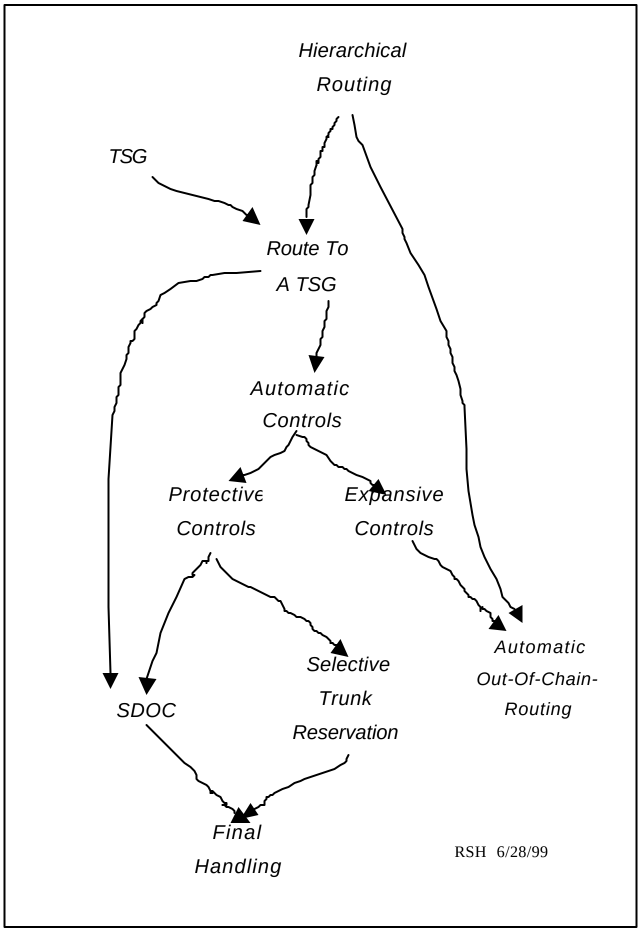
Natural disasters are another complicating factor within the network of switching systems. A common reaction to a disaster is to call your relatives to determine their status. When everyone does this a focused traffic load aimed at switches in the disaster area occurs. This is a kind of peak traffic that the network engineers cannot foresee, and hence frequently causes overload and control equipment congestion.

Many of these concepts: a variety of nodes, infeasibility of directly connecting every switch, and shared resources can be found within the surface transportation network as well as the telephone network. Each of the patterns in this language presents one or more examples from this other network to illustrate the principles discussed by the pattern. These examples also help show the power of these patterns in that they are not limited solely to an electronic communications network.

Language Map

The following diagram shows the relationships between the patterns in this language. The diagram shows patterns that enhance the solutions of other patterns, resolve previously unresolved forces in a pattern, or take advantage of an earlier pattern to provide some new system capability. In the example to the right, pattern *B* refines pattern *A*, helping to solve unresolved forces or new problems that *A* introduced.





The Patterns:

Trunk SubGroup (TSG)

Intent: Group trunks by common properties so that appropriate ones are easy to find.

Problem: From the myriad of different trunks in an office, how do I select the one for this particular call?

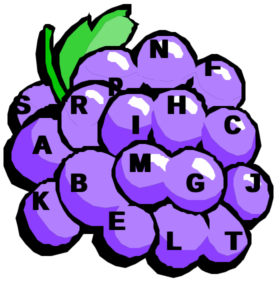
Context: Because of the heterogeneous nature of the switching systems in the telephone network, each is capable of performing call setup signaling with only a subset of the available protocols. Each of these protocols has different advantages and disadvantages for each switch that uses it. Each trunk has attributes related to the protocol used upon it. In addition to this each trunk will have a variety of fixed attributes such as endpoints and targeted traffic.

Forces: A switching network element will have many, many trunks, sometimes up to 100’s of thousands. Each of these has several different attributes that can be used to describe them. These attributes are of interest to the routing process so that appropriate classes of service can be allocated to each call traversing the network. Some of the example attributes are:

> Destination, Translation domain, switching domain, traffic use, far end, INWATS, incoming screening class, incoming traffic separation class, directionality, transmission delay, signaling characteristics. [MG]

The system cannot just pick any trunk from the pool of available trunks; it must have the right attributes. This requires that some record be kept of the attributes on a per trunk basis. These records can be used by the routing selection functions to locate the correct one to use.

Solution: When initially engineering the office group trunks into "trunk subgroups" (TSG) based upon common properties. These properties will allow trunk selection (or in the case of Network Management trunk category selection) to be easier since it will be easier to identify similar trunks at route selection time.



Rationale: This helps two aspects of switch application software: Network Management and Call Processing. Network Management is helped because finer divisions allow finer degrees of control. Call processing is allowed to select services by allowing calls to be routed to a particular type of facility.

Resulting Context: Trunk resources with specific attributes can be identified quickly because they are within the same TSG. This simplifies selection and allows the system to *Route to a TSG* at execution time.

Highway Example: Associate properties with each highway, such as maximum speed, how direct they are, tolls, etc. These can be used during route planning to provide the desired type of travel experience. Many modern trip assistant software packages actually allow the user to select which categories of roads that they will travel on. This allows the user to find routes with the properties that they desire.

Hierarchical Routing

Intent: Define a fixed way to route calls to neighboring switches

Problem: How do you design your network so that even unsophisticated switching systems can route calls through it automatically.

Context: You can't connect every office to every other. The method has to be easy enough for pre-electronic switching equipment to implement in hardware or with simple wiring changes. You want to design the network to make optimal use of its links and of its common resources.

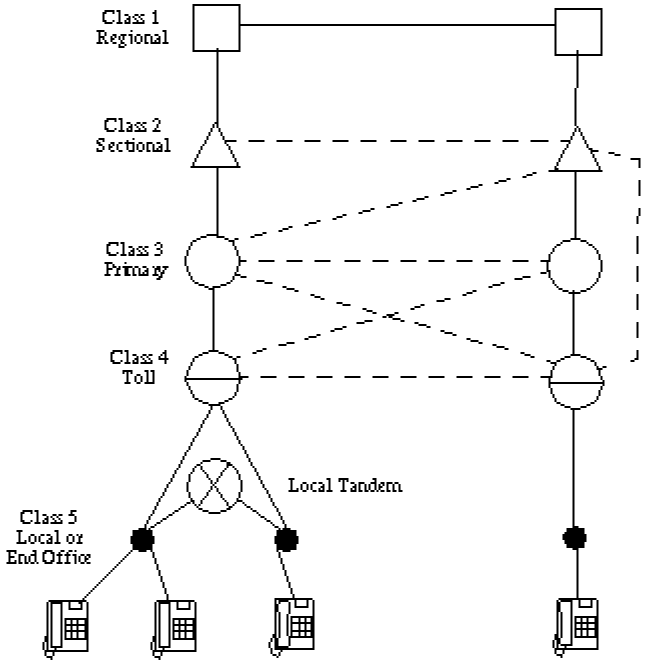
Forces: If you directly connect every node in a network you have n(n-1)/2 links. From experience we know that there is much expense associated with managing large numbers of links.

The more systems that must process a call, the longer it takes for the call to be established.

Solution: Identify each switch by a class number or level. In the telephone network five levels are used. Class 5 offices are end offices to connect to customer telephone sets. Class 4 offices are called toll offices, class 3's are primary centers, class 2's are sectional offices and class 1's are regional centers. Sometimes local tandem offices connect class 5 offices without the call requiring a class 4 office.

Route each call as low as possible. If the calling and called party are on the same switch, route the call directly. If they are not, go up the hierarchy from the calling party's class 5 office to the lowest switch (highest class number) that can connect the calling party's class 5 office with the called party's class 5. For example the call may be passed through the local office to the toll office to another toll office and back down to the called parties local, class 5 office.

Engineer some links between offices of different classes to provide some shortcuts within the hierarchy.



The assignment of a class number to a switch is something that is done by the network engineers as they are creating the network or allocating work to switching systems. The routes between switches that the network engineers design are accessed automatically during call creation to advance the call.

Rationale: A regularity has been added to the telephone network so that even systems that do not have computing power as we know it can route calls. The hierarchy provides that the higher the switch is in the hierarchy (the lower the class number) the fewer there need to be in a network.

Resulting Context: A large number of ramifications upon system design are introduced by using a hierarchical routing method, such as varying feature sets. These are not generally germane to this discussion of traffic congestion, except in the differences between lines and trunks that was made earlier. Lines connect telephone sets with Class 5 offices. Trunks connect switching systems. These patterns are primarily concerned with trunk traffic congestion.

Highway Example: The system in the USA of township roads, county highways, state highways, US highways and interstate highways. At each level there are fewer roads than at the next level, and each is designed to handle more traffic than the previous level.

Route To A TSG

Context: Trunks with similar properties have been grouped into *Trunk Sub Groups* during office engineering. Within a toll or tandem switch it is rarely required that a specific trunk be used for a particular telephone call. Generally the call must be forwarded from one switch to another by any available route. There are usually restrictions upon the types of trunks that are to be used when routing a call. For example the number of satellite links might be limited (due to the time delay from the great distances involved).

Problem: How granular should the "routing" function of a switch be?

Forces: Assume there are M routes between two switches and each route has N trunks. To identify a route all the way down to an idle trunk will take O(M*N) time.

To identify the appropriate route from among the M potential routes only takes M time. Then to identify the idle trunk within that route takes at most N time. Thus to break the route identification into two parts, one at the route level and the other at the trunk level takes only O(M+N) time.

Different algorithms might be more efficient for identifying routes and trunks. The deeper into the routing that you have to go to find the answer the more time and memory that you need. Routing to the trunk level requires the most levels. Routing to the trunk group requires fewer levels.

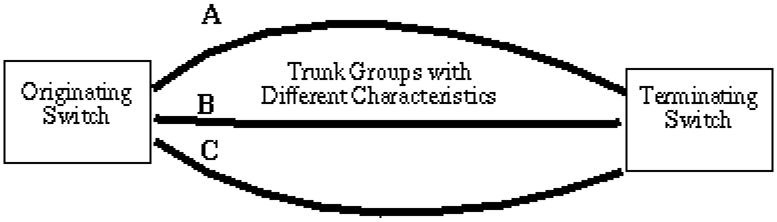
There are benefits to using the same algorithm to find both the route and the idle trunk, mostly in the maintenance area. Switching system performance is frequently more valued than its ease of maintenance.

There can be efficiencies if the problem being solved can be made smaller and optimal algorithms used for each sub-problem.

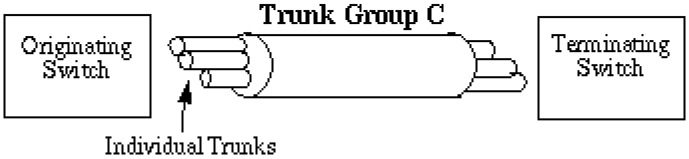
Solution: Routing a call across a network should occur in two steps. First, identify a route between the endpoints. This can be determined by examining the *Trunk Subgroups*. From there the selection of an individual trunk can occur without the overhead of the route selection process.

Selection of an idle trunk is a different problem than selection of a route between switches. Any idle trunk within a group will suffice, once the route has been selected.

Route only down to the trunk group.



Then use the most efficient method to get to an individual trunk.



Rationale: Using the most efficient method to select the route and using a different efficient algorithm to select the trunk within the route optimizes memory and execution efficiency of the trunk selection.

This pattern should be applicable to any context in which you have several "pipes" going from A to B with possibly different characteristics.

Resulting Context: A system where the most efficient algorithm is used for each part of the routing/trunk selection problem, even though that means that multiple algorithms are used. Pick the pipe that goes to the right place and then identify which flow within the pipe.

Let trunk group selection identify the major freeways to be taken, but leave the individual lane selection for more efficient (time and memory) methods.

Highway Example: Someone offering guidance about a route to take between cities doesn't also specify which lane to be in for the entire trip. Suggestions related to particular highway features might be offered, these are like the guidance offered by the different properties that were used to create the various *TSGs*.

Automatic Controls

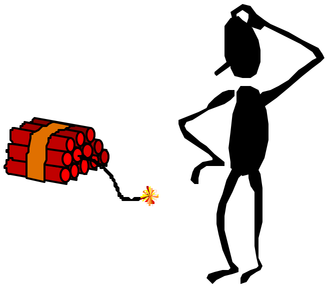
Intent: When conditions dictate, the switch should automatically institute changes to normal behaviour to handle extreme conditions.

Problem: People are slow, congestion problems grow rapidly

Context: *Minimize Human Intervention* [ACGH+] is in use, saying that the system should be able to take care of itself. The system is capable of computation. This eliminates many electro-mechanical switching systems from using this pattern.

Forces: People are slow and reluctant to respond. They look away, they miss things and frequently they overreact. Computers are good at deciding things if they've been prepared in advance for a given scenario. Most congestion situations are similar. They might be from a different source, but are basically more work than can be done in a given time.

Solution: The system should apply controls automatically. And be able to remove controls automatically as well. The system should allow humans to intervene (*People Know Best* [ACGH+]), but it should not rely upon them to intervene.



Rationale: The human element is eliminated, in keeping with *Minimize Human Intervention*.

Resulting Context: Having stated that the system should apply controls automatically, the actual controls need to be defined. Automatically applying controls is effective in both the processor and peripheral congestion areas. Some of the controls that can be applied are more applicable to one area than another (e.g. *Expansive Automatic Controls*), but overlap does exist as you will see.

Highway Example: There are a number of examples of automatic controls within the USA highway network. Frequently the highways have embedded sensors that will identify potential congestion and display it automatically to alert motorists that congestion appears ahead. Sometimes an automated radio broadcasts on a certain frequency to alert motorists. Sometimes the highways will automatically reconfigure themselves through reversible lanes that change to accommodate rush hour traffic flows.

Protective Automatic Controls

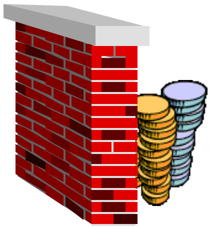
Intent: Protect the system from too much work by restricting what work it will allow into the system.

Context: The system is capable of applying *Automatic Controls*. The metric being watched is throughput – the amount of traffic (calls) that will be successfully processed during a period of time, and congestion is occurring for the finite resources within the system. For example, during an event where many customers are dialing in simultaneously and there is a shortage of the peripheral equipment that processes touch-tones.

Problem: What actions should the switch take when the resource being congested has a finite limit?

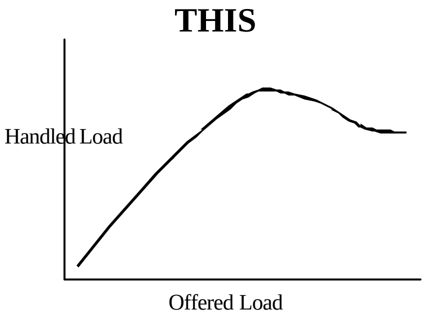
Forces: If the system takes on all the work that is being offered for the finite resource, requests will have to be queued for service when resources are freed. This takes work to queue. And if the congestion is of more than brief duration queue lengths will increase and yield unacceptable delays. The nature of the traffic then also comes into play. Human callers get impatient, hang up and retry their call. This results in requests being queued that are no longer valid as well as additional requests for service.

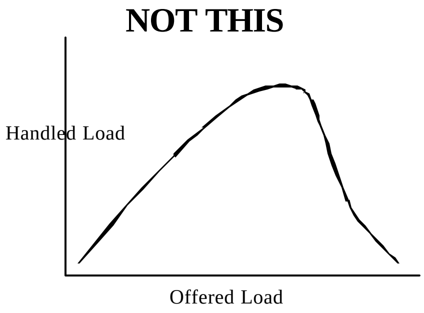
Solution: Impose *Automatic Controls* that restrict how much work the system accepts. Sometimes there are things that the system can do that will safely protect itself from too much traffic. These should be invoked when the congestion is in elements that have finite limits and no idle capacity that can be used.



Rationale: There are three basic responses that a system can have with confronted with congestion. The first is to shrink within itself, taking whatever actions are necessary to be protected from the onslaught of too much traffic. The second is to open itself up to the traffic, taking on as much as it can and shedding work that is unnecessary. The third is to not do anything, which will generally lead the system into instability.

A typical response to too much work is for the system to begin thrashing. The system takes on more than it can handle, response times increase and the problems just magnify. This causes throughput to decrease greatly. By protecting itself the system will have a lower throughput than it did before the automatic control started, but the drop off will not be as severe as if nothing had been done.





Resulting Context: Typically processor real time is considered a finite resource and Protective Controls are needed to deal with congestion for it.

Related Patterns: This pattern applies when the resource is finite. If there are ways of expanding the range of possibilities – of increasing the amount of the resource then apply an *Expansive Automatic Control*.

Within this pattern language, resources are things other than processor real-time. Gerard Meszaros [Mesz] has related patterns for the real-time case beginning with the pattern *Shed Load*.

Highway Example: Metered freeway entrance ramps are examples of *Protective Automatic Controls*. When the finite resource of highway capacity becomes congested it is possible to automatically restrict the rate at which new traffic enters by putting stoplights at the top of the entrance ramp. At predetermined intervals the stoplight will turn green and will allow one request for service (car) onto the freeway. This limits the amount of new work that is added to the system, at the risk of backing traffic up at the entrance ramp.

Selective DOC

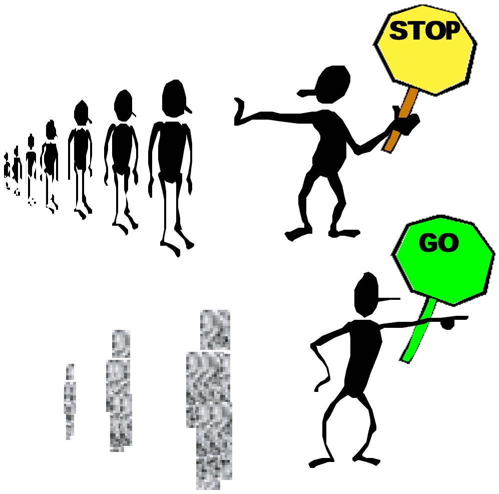
Intent: Tell connected systems parts of our system are overloaded.

Context: The system is able to apply *Automatic Controls*. Trunk resources have been grouped into different *Trunk Sub Groups* by their attributes, and routing is done to the trunk sub group level (*Route to a TSG*). *Dynamic Overload Control* (*DOC*), a different form of *Protective Automatic Control* shuts off all traffic to a switch.

Problem: How can certain kinds of traffic be allowed? Is there some work that the system is capable of handling even though it cannot do some work of different kinds? Can the system process any traffic? Are there only certain types that are congested at the moment?

Forces: Frequently congestion is localized. A certain originating or terminating node is congested (as during a natural disaster). Or a certain type of resource is much overloaded, possibly because of a failure somewhere else in the network. Even in periods of processor overload, or traffic congestion, there will be some idle resources that can be used to allow traffic through.

Solution: Request of systems that traffic is routed to that they limit the amount of traffic that they send, but only for the particular properties (as identified by *Trunk SubGroup*) that are in congestion.



Highway Example: High occupancy vehicle lanes frequently allow their types of traffic avoid congestion on the main thoroughfare through the use of dedicated exit ramps. Since they are dedicated they are much less likely to be congested than the main roadway.

Selective Trunk Reservation

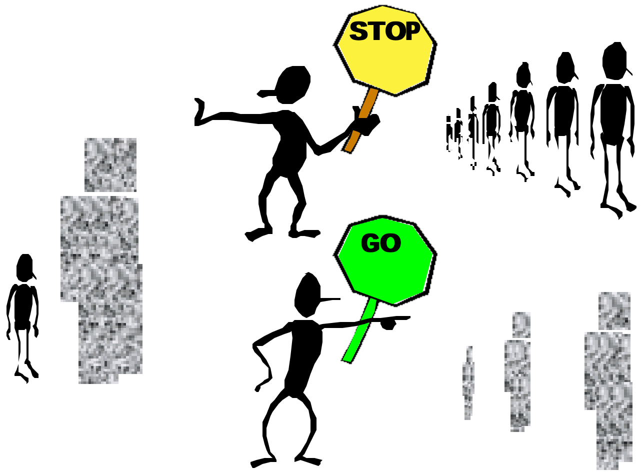
Intent: Deny incoming traffic on *TSG*s that have few idle trunks during periods of congestion.

Context: The system is designed to automatically apply *Protective Automatic Controls*. Actions will be taken that will allow it to protect itself, to maximize the throughput through this individual system. The system has too much work to do and not enough of some peripheral equipment resources.

Problem: How should the system handle requests to the most busy *Trunk SubGroups*?

Forces: Calls that must be retried several times to be completed tie up resources longer than calls that complete on the first attempt.

Solution: The traffic most likely to succeed on *TSG*s that have many idle trunks. To protect the system during congestion, favor those calls over the ones that arrive on busy *TSG*s. Mark certain outgoing trunks in such a way that calls to them are allowed to be processed. Calls to the unmarked trunks will be blocked.



Resulting Context: A system with Selective Trunk Reservation will spend more time on work that will actually succeed rather than work that will only delay other work.

Highway Example: On roadways that have less traffic, provide longer green traffic signals than on the congested roadways. This favors keeping the less busy roadway flowing smoothly. The other roadway is already congested – it won't make much difference. This is very similar to "Late trains get later".

Expansive Automatic Controls

Intent: Protect the system from too much work/traffic by providing new ways to do the work.

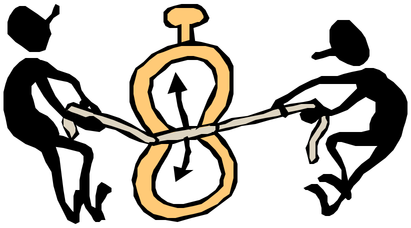
Context: *Automatic Controls* can be applied. In this case the metric being watched is the number of calls that reach their destination. This is a more network-centric metric than that being watched in *Protective Automatic Controls* (which is system throughput). Resources within the network are not totally finite. New links can be created to expand it's the network capacity.

Problem: How can we both protect a system from the effects of overload and at the same time not reduce overall throughput?

Forces: If purely *Protective Automatic Controls* are used when confronted with congestion, each switch will shrink back upon itself and network throughput will decrease, which is opposite the objectives of this pattern. . If some way to expand the range of possibilities available to a system were possible then overall network throughput might not decrease. Are there alternate ways that a system can deal with its workload? Ways that are normally not used? Can any of these ways be put in? If used for ordinary traffic they don't expand it during time of network congestion.

Solution: Design relief mechanisms into the network that can be accessed only in case of congestion. Provide new ways for the switch to do its work that a) has a higher probability of succeeding or b) use fewer switch resources.

Impose automatic controls that provide new ways to do the same work. This allows excess work to be taken on and the congestion lessened. It is like providing an escape valve.



Rationale: There are three basic responses that a system can have with confronted with congestion. The first is to shrink within itself, taking whatever action is necessary to protect itself from the onslaught of too much traffic. The second is to open itself up to the traffic, taking on as much as it can and shedding work that is unnecessary. The third is to not do anything, which will generally lead the system into instability.

Expansive Automatic Controls result in the network throughput being managed

Resulting Context: *Expansive Automatic Controls* provide new ways for the system (element in a network) to do its work. This requires work at the network level rather than the level of individual systems.

In some scheduling regimes it is possible to defer some internal maintenance work during peak traffic times. This expands the amount of real time that is available to perform revenue-generating work. It also allows this pattern to be applied to processor capacity.

Related Patterns: *Protective Automatic Controls* deals with finitely bounded resources. This pattern deals with resources that can be expanded.

Highway Example: An example of an expansive control is when the police direct traffic over side streets, possibly in violation of posted regulations, to avoid an accident scene. They have the authority to take this action, but they seldom exercise it. Another example is using all lanes of a highway in one direction to allow a higher number of people to leave an area threatened by a natural disaster such as a hurricane.

Automatic Out-of-chain Routing

Intent: During overloaded periods, allow new routes within a hierarchical network.

Context: The system has the capability to apply *Automatic Expansive Control*. The network employs *Hierarchical Routing*. The network is being engineered to maximize throughput.

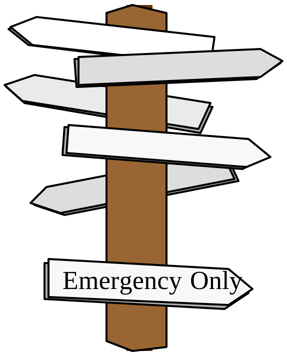
Problem: How can the effects of congestion on trunks to the next office in the *Hierarchical Route* be reduced?

Forces: The routing chains of a hierarchical network are quite rigid. And the network designers can look at it in advance during the engineering phase and calculate the total possible throughput. These limits will be exceeded sometime.

Traffic can be restricted through a *Protective Automatic Control*, but that will decrease throughput.

A *hierarchical network* does not connect every system to every other. There are many possible routings between switches that are not part of the standard hierarchy of the network. New routes between systems might be nothing more than a jumper cable in another office (i.e. an alternate route from A to C might really be an A to B route permanently connected to a B to C route), they might not need to be a direct cross-country route.

Solution: During periods of traffic congestion, allow the system to use routes that are not normally used. They are “out-of-chain”. Since these routes are not usually used they will not be congested.



Rationale: Providing a way to use some routes only during an emergency is a way of improving upon *Hierarchical Routing* to expand the options available to the system. This is an *Expansive Automatic Control*.

Highway Example: The examples listed above in *Expansive Automatic Controls* are examples of Automatic Out-Of-Chain Routing. They legalize ways of getting traffic moving that are normally not allowed. In some areas, at particularly congested exit ramps, there will be signs posted that allow shoulder riding between certain hours. This allows the motorist to do something that is normally illegal that helps relieve the congestion.

Final Handling

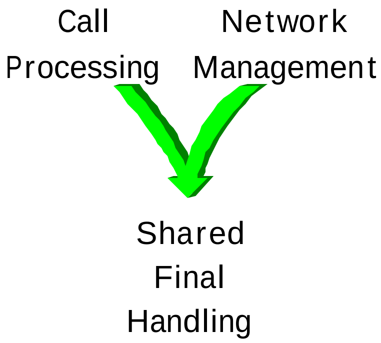
Intent: Gracefully tear down a call that cannot complete.

Context: Normally the taking down of a call connection after hanging up is a function of call processing programs after one of the parties hangs up. Sometimes a call cannot complete for some internal reason, such as a *SDOC* control. These calls and all the resources that it has tied up need to be released.

Problem: Should network management release the resources for any calls it terminates abnormally?

Forces: If they do they can be sure to collect any data that they might need for analysis. But the steps involved in releasing these resources are generally just like the releasing of a normal call when one party hangs up. But call processing will have extra work to do in the form of the forward connections that network management didn't allow to be completed. The choices are to do it separately or use the same code for both.

Solution: Integrate the release of resource for internally terminated calls with the usual call processing release of resources.



Rationale: In this way there are few possibilities of forgetting to release some part, saving development and testing time. This saves development because it needs to only be written once. It also saves memory space. In a system that is designed to be highly available, the benefits of single development and maintenance of only one code are great.

Resulting Context: Data should be collected when calls are terminated abnormally. This data needs to be presented to human network managers so that they can make changes to strategy or network topology.

Highway Example: On some freeways there are emergency pull-off locations where it is safe to stop and report an accident or vehicle problem without obstructing the normal flow of traffic. After the problem is reported or corrected the normal lanes are used to resume the trip up to a normal exit. These lanes do not provide special exit ramps. They provide the ability to perform some special actions and then the normal exit ramp is used.

Acknowledgments:

The authors acknowledge the many very useful comments of the PLoP shepherd David DeLano.

Dana Garoutte and Andrew Peck of Lucent Technologies originally wrote *Route to A TSG*. It has been revised here to fit within the context of this language.

Appendix: Related Patterns

Pattern Name	*Pattern Intent*
Dynamic Overload Control (DOC)	Tell connected systems we're in trouble!
Minimize Human Intervention	Give machine enough smarts to not require human intervention
People Know Best	Assume humans know more than the machine.

Glossary:

Call Processing: The switching system application that connects telephone calls from *Calling party* to *Called Party.*

Congestion: When the system receives more requests for service than it has peripheral resources to handle them.

Control (noun): Changes to normal call processing instituted during congestion or overload to allow the system to either handle additional traffic, or to protect itself from insanity.

DOC: Dynamic Overload Control. A method of alerting connected switches to request a reduction in traffic load from that switch.

Network Management: Within the context presented in this paper, Network Management refers to those capabilities used to monitor and control, in real time, a network of telecommunications resources.

Overload: When the system has more work for the processor than it can handle in the available processor resources.

Traffic: Telephone calls, or to be more generic, any requests for service, such as read requests to a tape driver.

Trunk: The logical connection of the switch to the physical line over which inter-switch traffic arrives.

References:

[ACGH+] Adams, M., J. Coplien, R. Gamoke, R. Hanmer, F. Keeve and K. Nicodemus. 1996. "Fault-Tolerant Telecommunication System Patterns" in **Pattern Languages of Program Design,** edited by J. M. Vlissides, J. O. Coplien and N. L. Kerth. Reading, MA: Addison-Wesley Publishing Co.

[GHHJ]: Green, T. V., D. G. Haenschke, B. H. Hornbach and C. E. Johnson. 1977. "No 4 ESS: Network Management and Traffic Administration." Bell System Technical Journal, vol. 56, no. 7, Sept, 1977: 1169-1202.

[MG] DiCarlo-Cottone, M. J., and E. Gomez. "Network management – Beating the Bottlenecks." Bell Labs Record, vol. 55, December, 1977; 305-309.

[Mesz] Meszaros, G. 1996. "A Pattern Language for Improving the Capacity of Reactive Systems" in **Pattern Languages of Program Design,** edited by J. M. Vlissides, J. O. Coplien and N. L. Kerth. Reading, MA: Addison-Wesley Publishing Co.