

Envisaging an Active Network Architecture

Presented by:

Jayakrishnan K Nair

The Vision behind Active Nets

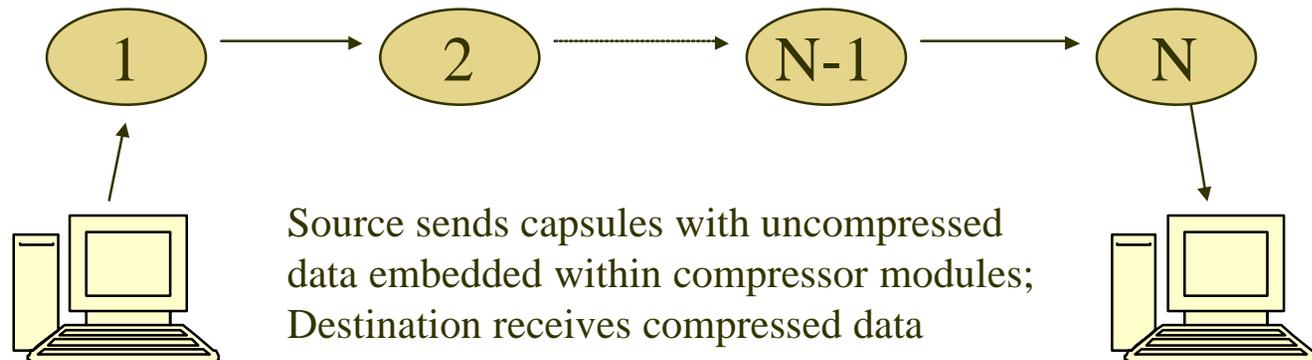
- The purport of Traditional Networks is merely to transport bits from source to destination.
- The data thus transported is opaque to the network.
- Since there is no semantic processing of data, the only computations performed are from a transport perspective, like CRC & checksum

On the other hand.....

- Active Networks allow users to inject customized programs into nodes of the network.
- Thus custom semantic processing on data is performed at the network during transit and thus data is modified into a more readily operable form

A Conceived Modus Operandi

- Packets can be replaced with active Capsules
- Capsules have code embedded in them so that they are executed at each node they traverse
- These mini-programs incorporate within them the user data also, a la in PostScript code
- Each node may have predefined program methods that may be dynamically invoked by the capsules



Advantages of Active Networks

- Unlike in the conventional networks, the nodes are not dormant to data processing, and hence their latent computational potential is utilized owing to a much better service throughput at destination.
- Portability of services across domains using highly adaptive protocols is maximized, yielding richer interactions and hence obviating the necessity for the exchange of fixed data formats
- Strategic juxtaposing of fine grained application-specific functions on the nodal graph is achievable.

Advantages of Active Networks

- Adds versatility and scalability to the system to enhance scope for new services, as the bottleneck of vendor-driven standardization is assuaged
- Provides a sagacious and clairvoyant next step to the emanation of cutting-edge active technologies that support encapsulation, transfer and safe execution of program fragments.
- Provides a viable and more beneficial alternative to ad hoc contrivances of firewalls, Web proxies, mobile proxies, multicast routers, video gateways et al, which need functionally active nodes

Applications of Interest

- Mobile Code need to travel from client to server for agent technologies, and in vice-versa for Web applets, in the traditional framework
- Active Networks eliminate this dichotomy by allowing applications to dispatch computation en route to the destination
- In this perspective, several application designers come to the fore, such as for Firewalls, Web Proxies, Mobile/Nomadic Gateways etc.

Firewalls

- Firewalls are security applications that run on interfaces between networks.
- They implement filters based on various fields in the packets like source/destination IP, service etc
- Using Active nets, the automation of application authentication is possible by injecting appropriate filter modules that set the permissions.

Web Proxies

- Dynamic updating of WWW pages in the web caches has been a challenging research topic
- Active Nets yield a promising solution
- Web proxy schemes can support active caches that dynamically execute programs that generate web pages on the fly, thus minimizing web accesses.

Mobile/ Nomadic Computing

- The bandwidth of a system's connection to the network could vary heavily depending on the changes in the type of connection (T1/ Dial-up etc)
- A nomadic router is able to adapt to this with the view to optimize performance from the available bandwidth, by varying cache size etc.
- Nomadic gateways are similar – they are located between networks of highly different bandwidths, like between wired and wireless networks
- Active networking can ameliorate the situation significantly, by using application-specific services such as file caching, image transcoding etc.

Myriad Novel Vistas

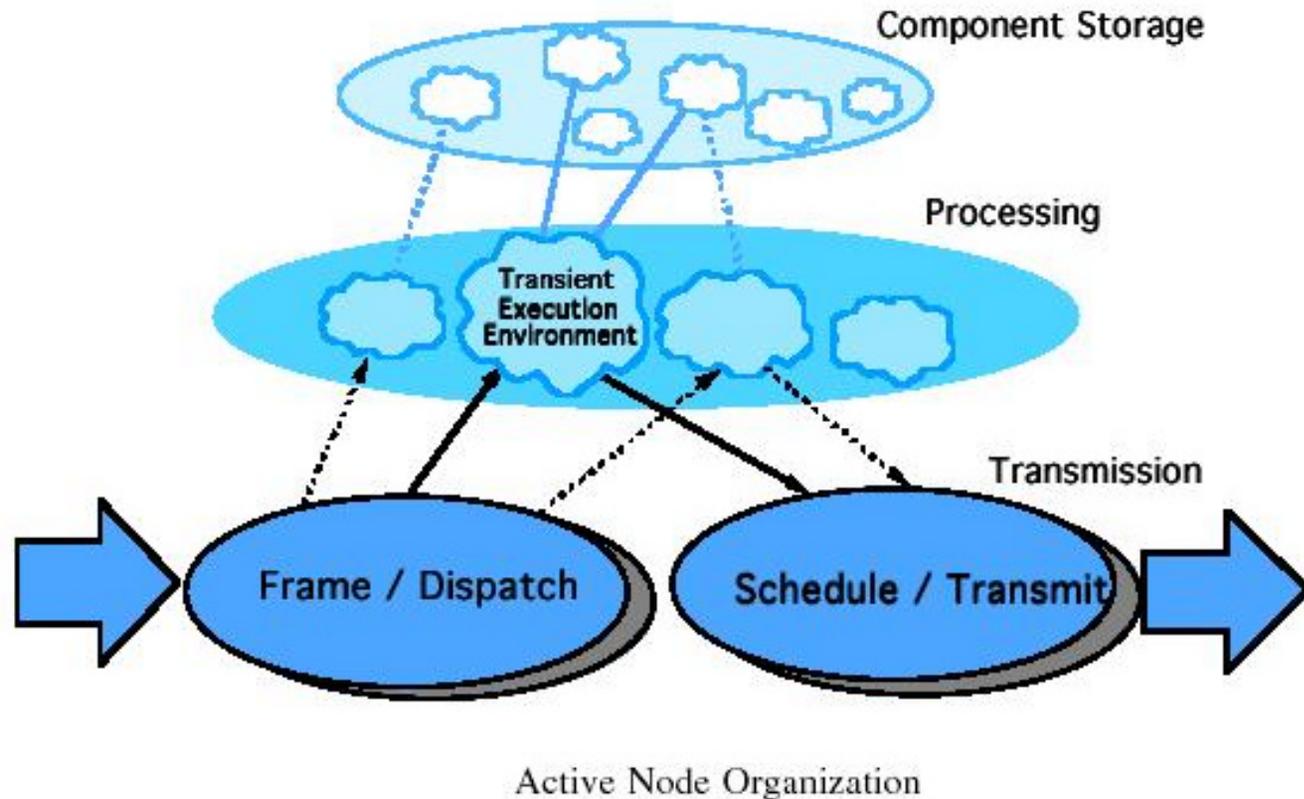
- Sophisticated Network-based services currently face bandwidth constraints, keeping them from realizing their true potential
- Active Network is the closest to a panacea we can get, as fusing data within the network can reduce bandwidth requirements
- Development of more flexible multi-point communication strategies than the current IP multicast service can be supported
- Enhanced Video Conferencing and Composite Image Viewing from distributed sensors are two examples

An Implementation Perspective

- One intuitively suggestive architectural design perspective is the usage of programmable switches
- Here, each node can be programmed to operate on the incoming packets after header removal.
- The header of the packet will identifies which among the available modules is to be run.
- In the case that there is no program associated with a particular header, the node passes the packet on just like a traditional passive node.
- Stringent operator authentication is required for this discrete approach, to obviate malevolent tampering.

An Integrated Design Approach

- An extreme case would be to have every message being embedded in a program capsule.



How it works

- When a capsule arrives at an active node, it first identifies the capsule boundaries using framing mechanisms in the link layer
- It then dispatches the contents to a transient execution environment
- The embedded code in the capsule contain primitive instructions, and can invoke external methods to access the resources in the router.
- The accessibility to such external objects, like routing tables, is determined using the concepts of Foundation Components, Active Storage and Extensibility.

Foundation Components

- These components are inherently present in each node, to implement external methods that provide controlled access to the router's resources
- Some of them reflect the API of the node's runtime environment to the embedded application
- Others provide a built-in class hierarchy for an access model to resources

Active Storage

- This is a method to preserve state information so that coherent packets can save computation by making use of the saved states from predecessors.
- Preplanned congregation of capsules at a node for a joint computation on a particular node, can be achieved by each capsule setting state information
- Until all required capsules arrive, the earlier capsules would lie dormant.
- Finally when the state in the active storage changes to signal end of hibernation for dormant capsules, they begin to process in unison

Extensibility

- This is an interesting approach for building a large programming environment block by block
- It is impractical to have large cumbersome code to be squeezed into individual capsules.
- However, it is possible to break the program in an object-oriented perspective.
- Thus, uniquely named classes and methods can be implanted in to the nodes by individual capsules as they traverse through them.
- After all the building blocks have arrived, the node will have the environment ready for execution.

Mobility using Capsule Programs

- Mobility and Portability issues stem from the need for versatility for the capsules to be transferred and executed across a range of platforms.
- Expressing the source code in high-level language enables it to be interpreted at the nodes.
- Another method is to adopt a platform-independent intermediate representation such as a byte-coded virtual instruction set.
- A less elegant method is to have each packet carrying multiple binary machine codes corresponding to each of the traversable platforms.

Safe and Efficient Execution

- A critical design consideration is to have safety and performance aspects apposite to each other.
- Safety stems from the need to restrict accessibility to critical resources, in order to obviate catastrophe.
- Efficiency is the requirement to meet the safety consideration with minimal performance overhead.
- Active Technologies can be leveraged to provide safe and efficient execution by restricting the operand scope and available primitive instructions.

Active Technologies to the fore

- Source Code – Languages such as Safe-TCL provide namespace closure, and prevents programs from straying beyond their execution environment
- Intermediate Code – An intermediate instruction set as used by Java inherently precludes the need for the interpreter to check certain cases, as in operand validation, thereby improving efficiency
- Binary Code – Directly executable machine code costs in mobility, but gives high performance gain.
- The instruction set and address-space must be restricted in the last case, to account for generic processing by any conforming hardware.

More on Active Technologies

- The safety aspect of using such direct binary code can be achieved by using a trust-worthy compiler that always confines program variables to a closed environment, which is verified at the node
- A set of smart sandboxed techniques, such as restricted address arithmetic and run-time support can work in tandem to achieve the same goal.
- Automatic regeneration or modification of source-code using on-the-fly compilation, along with sandboxing, can further make the safety-efficiency balance more apposite.

Interoperability Aspect

- The active network will have to share resources with the administrative domains, and hence the need for efficient interoperability is paramount
- A minimalistic resource model is sufficient for most capsules.
- Transient bandwidth, Processing Capacity, Transient Storage and Active Storage are some of the most relevant physical resource indices needed to be abstracted into the capsules.

Repercussion on OSI model

- The active network model points out the need to address the lacunae of the layered reference model:
 - Upper-layer services are possible only on an end-to-end basis, and is opaque to the network
 - Services at or below network layer are application-independent
 - Tunneling of networks causing recursion is overlooked

Conclusion

- By implementing processing capabilities in networks, which has hitherto been only passive forwarding medium, it is possible to elegantly leverage current technologies to derive higher performance and QoS at low overheads
- The “Chicken and Egg” problem raises here as the vendors wait for this late technology to become the “*de facto* standard” whereas that is possible only if large-scale upgrading is done by the vendors !
- However, the scope opened by active networks is too enticing to be sidelined, making the synergetic development of a research ActiveNet expedient.

Thank You Very Much !